A Comparative Study of Software Development Practices in the Context of an Emerging Country

Partha Chakraborty[a], Khalid Hasan[a], Rifat Shahriyar[a], Anindya Iqbal[a], and Gias Uddin[b]

[a] *Bangladesh University of Engineering and Technology and* [b] *University of Calgary*

Abstract

Software development is complicated with rapidly changing requirements, techniques, processes and the involvement of diverse stakeholders. A study on the software development practice in a country can reveal the challenges and scopes of improvement for countries of similar types. There are several studies on the software industry of developed countries (e.g., European countries, Canada, New Zealand, etc.). To the best of our knowledge, no recent study explores the software development practices and challenges in an emerging country. We focus on understanding the challenges and opportunities within the software companies of Bangladesh, which is an emerging developing country with a large group of skilled software practitioners. We have two objectives. First, we aim to understand the methods and practices adopted and challenges faced by the software companies in an emerging country like Bangladesh. Second, we aim to understand whether and how the development practices and methods in Bangladesh differ from other countries. The comparison informs us of the uniqueness of software industries in emerging countries like Bangladesh. Our study has four phases. First, we conduct a series of semi-structured interviews with eight individual practitioners from four software companies. The purpose is to understand the overall development practices and methods. Second, we use the insights gained from the first phase to design a survey. A total of 137 software practitioners from a diverse set of companies responded to the survey. Third, we analyze the survey responses to learn about the development practices in Bangladesh. Fourth, we compare our findings against findings previously reported for other countries (e.g., European countries, Canada, New Zealand, etc.). The major findings on the companies of Bangladesh software industry are: i) Most companies follow agile methodology and give implementation-related activities a high priority, ii) Most companies develop web-based software, iii) Most companies use less automated testing and deployment tools, but iv) Several companies seriously consider the security, performance, and scalability of their software. The development practices in Bangladesh are mostly similar to other developing countries (e.g., Malaysia), but differ from those in developed countries (e.g., Japan, Canada, Europe, etc.). For example, due to the mostly outsourcing nature of work carried in Bangladesh, overall system design is not as highly practiced. Automated testing is also not as widely practiced in Bangladesh as the developed countries, partly due to the difficulty of testing web-based systems that are mostly developed in Bangladesh. The findings can guide software practitioners to improve development practices in emerging countries and software providers with new tool support to assist in the process. In particular, more emphasis can be placed upon promoting overall system design and automated testing in the software industries of emerging countries.

Keywords: Survey, Software Development Practices and Challenges, Emerging Countries

1. Introduction

A study on the software development practice in a country can reveal the challenges and scopes of improvement for software industries of similar countries. There are several studies on the software industry of countries that focused on understanding the software development practices in developed countries and/or countries with matured software industry (e.g., Japan, Canada) [1, 2, 3, 4]. However, the software industry of a developed country differs from that of a developing country in many aspects. Easy migration opportunities of the developers cause constant scarcity of experienced people in developing countries.

Still, due to the abundance of Computer Science graduates, the software industry of some developing countries are continuing to emerge by catering to the majority of the local market as well as expanding in the global market.

A recent study focused on the software development ecosystems in Malaysia [5]. However, the study did not consider the diverse development practices (e.g, security, scalability in development practices). There is also no study on a systematic comparison of the software development practices in the software industries between the developing and developed countries. In this paper, we focus on understanding the software development practices in an emerging software industry by analyzing those practices in another developing country, Bangladesh. Like Malaysia, Bangladesh is also a rapidly growing economy with 160 million population. IT sector is considered a priority sector in Bangladesh over the last decade. The software development industry dominates this sector. According to the Bangladesh Association of Software and Information Services (BASIS), 1100+ software companies operate in Bangladesh, where around 40% have a global business. The foreign revenue earned by the industry is over 800 Million USD [6]. Therefore, an understanding of the development practices in Bangladesh and a comparison of such practices with other countries can inform us of the issues and characteristic of an emerging software industry.

Our study has four steps. First, we conduct a series of semi-structured interviews of eight software professionals from four leading software companies in Bangladesh. We corroborate the interview findings with findings from similar studies of other countries like Canada, Turkey, Netherlands, New Zealand, etc. [1, 2, 3, 4]. The purpose is to gain an overview of the overall development practices and challenges. Second, we design a survey based on the insights gained from the interviews. A total of 137 software practitioners from diverse software companies in Bangladesh responded to the survey. Third, we analyzed the survey responses to understand the software development practices and challenges in Bangladesh. Fourth, we compare our findings against findings from other countries. We answer two research questions: (RQ1) What are the software development practices in an emerging country like Bangladesh?

(RQ2) How do the development practices in Bangladesh differ from other countries? We report the practices (RQ1) and the comparisons (RQ2) along four dimensions (D) as follows. The four dimensions were previously used in similar country-based previous studies, which thus helped us to compare our findings against their findings.

D1. Software development methodologies used. We investigate the development approaches, methodologies, and requirements analysis processes. We find that the software companies in Bangladesh mostly follow the agile methodology. In comparison with the other countries, Bangladeshi software companies generally spend more time on the implementation stage of the development. Technologically advanced countries spend more on system design.

D2. Software tools and techniques used. We determine the trending technologies like technology platforms, programming languages, frameworks, etc. We find that web-based software services are prevalent in the Bangladeshi software market and JavaScript is mostly used language for web development. The degree of technologies and tools usage varies from the developed countries due to the availability of experienced developers, budget, etc.

D3. Software testing and devops practices used. We explore the present situation of testing and deployment practices adopted by the software firms in Bangladesh by asking questions about testing and deployment tools, test automation level, version control system, etc. We find that the usage of test automation and deployment tools is not widespread in the Bangladeshi industry. We also see that compared to developed countries, the test automation tool adoption is inadequate.

D4. Performance and security measures used. We analyze how software companies secure and maintain their code and what practices are followed to ensure performance and scalability. The responses show that standards are followed for security assurance, tools are used for performance testing, and scalability is mostly ensured by using cloud services. However, the companies in Bangladesh lags behind in the area of automated performance testing and containerization that might ensure resource-optimized scalability.

The findings can help software practitioners to improve development practices in Bangladesh, which then can be applied to other developing countries with similar characteristics. The comparison with the developed countries shows the particular avenues for improvement in the emerging countries. We also notice the popularity of using cloud services across the software industries in Bangladesh. While this is not surprising, it nevertheless can motivate cloud providers to invest in usable and readily available automated software design and testing tools into the cloud to cater to the needs of emerging software industries. Finally, our study catalogs the technologies and tools currently being used in Bangladesh's software firms, which could be useful for current students to prepare themselves for a career in this emerging software industry.

Replication package of this study is available at `https://git.io/JLtxI`.

Paper Organization. Section 2 presents the related work to our study. Section 3 describes the background of our study and the data collection procedure. Section 4 answers the two research questions. Section 5 reports challenges and opportunities as well as analysis by several dimensions of the findings. Section 6 discusses the implications of our findings. Section 7 discusses the threats to validity. Section 8 concludes the paper.

2. Related Works

This section presents the related works that focus on software development practices and processes globally and in specific countries and regions.

2.1. Studies of Development Practices and Processes

Cusumano et al. [7] have conducted a global survey on a completed software engineering project to identify software engineering practices. They have found that detailed architectural design and documentation is a common practice worldwide except the USA. In the USA, only 32% of projects used detailed design specifications. One of the interesting findings of their study is the completeness of design before coding negatively correlates with the number of defects.

AlSubaihin et al. [8] have identified the influence of the app store on software engineering practices. They have found that the perception of quality is slightly different among app store developers. App Store developers gave more priority to user rating than the traditional metrics like code quality and documentation when measuring software quality.

To identify the state of the practices in start-up companies, Klotins et al. [9] have conducted a study on start-up companies. They found that start-ups apply market-driven requirements engineering instead of the standard software engineering requirement engineering. However, the applied requirements engineering practices are often rudimentary and lack alignment with other knowledge areas (such as design).

2.2. Related Region-specific Studies

In 2012, Vonken et al. [3] surveyed Dutch software producing organizations to determine whether there is a gap between the current state of the practice and state of the art in software engineering. From 99 respondents, they extracted 22 interesting observations. These observations mark insights into the development process that they found unusual or surprising, at least from an academic perspective. This unusualness could either stem from certain principles being applied less or more frequently than expected or from unexpected correlations observed between factors.

The survey conducted by Garousi et al. [2] studies Turkey's software practices to characterize and understand the state of its SE practices. The military and defense software sectors are quite prominent in Turkey, especially in the capital Ankara region, and many SE practitioners work for those companies. 54% of the participants reported not using any software size measurement methods, while 33% mentioned that they had measured lines of code (LOC). In terms of effort, after the development phase (on average, 31% of overall project effort), software testing, requirements, design, and maintenance phases come next and have similar average values (14%, 12%,12%, and 11% respectively). Respondents experience the most challenge in the requirements phase. As a rather old but still widely used life-cycle model, the Waterfall is the model that more than half of the respondents (53%) use. The next most preferred life-cycle models are incremental and Agile development models with 38% and 34% usage rates, respectively. The Waterfall and Agile methodologies have slight negative correlations, denoting that if one is used in a company, the other will less likely to be used. A recent survey conducted by Wang et al. [4] in 2018 shows that New Zealand professionals use similar methodologies as professionals in other countries. Key findings of their study are, (1) popular programming language in the New Zealand software industry does not match with the worldwide ranking of popular languages, and (2) most of the time in SDLC is spent on implementation-related activities rather than analysis and design.

In another study, Groves et al. [10] reported that the New Zealand software industry pays particular attention to requirements gathering. They surveyed a selection of software companies with a general questionnaire and then conducted in-depth interviews with four companies. They found a clear difference in the testing phase between large and small software companies. Their finding is larger companies pay more attention to testing than smaller companies.

The study conducted by Sison et al. [11] presents exploratory survey and case study results on software practices of some software firms in five ASEAN countries (Malaysia, Philippines, Singapore, Thailand, and Vietnam). They found that most of the firms in that region do not follow the standard procedure for SQA. In a study focusing on the test practices in Canadian firms, Garousi et al. [1] found that the number of passing user acceptance tests and the number of defects found per day are considered the most important quality assurance metrics in Canadian firms. They compared their result to a previous study and showed that Canadian firms are giving more importance to testing related training than in the past.

Baharom et al. [5] had conducted a study on Malaysian software firms to find the effectiveness of standard practices. They found that alpha and beta testing was hardly implemented in software firms. Another interesting finding of their study is that most Malaysian firms emphasize implementation; only a negligible number of companies spend more than 20% of the effort in planning and design. Zafar et al. [12] have surveyed why Pakistani software firms do not follow the standard requirement engineering process. They have found multiple factors contributing to the issue, such as lack of budget, lack of time, lack of dedicated team, etc. However, the most prevalent issue is lack of budget; more than 60% of their respondents have responded that the standard requirement engineering process was not followed due to scarcity of budget.

Based on a 200 participants survey, Hussain et al. [13] concluded that computer science undergraduate education system in Bangladesh leaves most of its graduates unprepared for the software industry. They suggested that updating the syllabus as part of the curriculum and including internships could help make graduates fit for the industry.

To identify the software engineering practices in Bangladesh, Rahim et al. [14] have surveyed 41 practitioners in the Bangladesh software industry. One of their interesting findings is the waterfall model is still popular among them. They found that 40% of respondents indicated that requirement analysis and prioritization are the most challenging software development process. In another survey focusing on testing practices in the Bangladesh software industry, Bhuiyan et al. [15] found that most companies do not follow any standard SQA techniques for their projects. The interesting fact is such malpractice does not hinder their progress; they reported that these firms are in the industry for 6.5 years on average. However, in another survey, Begum et al. [16] found that 47.5% of respondents follow standard SQA techniques in their projects.

3. Study Setup

Our study has four steps as follows:

- **Step 1 - Interview (Section 3.1).** We conduct a series of semi-structured interviews to understand the current development practices in Bangladesh and to understand how such practices could differ from other countries.

- **Step 2 - Survey (Section 3.2).** We design a survey questions to get deep insights on the insights collected from the interviews.
- **Steps 3, 4 - Data Analysis to answer RQ1 and RQ2 (Section 3.3).** We analyze the survey responses to answer RQ1 (understand development practices in Bangladesh) and RQ2 (compare the practices against other countries).

3.1. Interview

The goal of the interview session was to prepare the survey questions. Eight individual participants from four leading software companies were interviewed. First we designed an initial list of survey questions in Google form by consulting previous studies in other countries like Canada, Turkey, Netherlands, New Zealand, etc. [1, 2, 3, 4]. Each participant was asked to identify ambiguities in the question. From the feedback of the interview session, we revised the questions. Each interview session lasted about half an hour. Interviewees were first asked to complete the survey. After completing the survey, we asked what he/she understood from the questions and what he/she meant by the answers. Throughout the interviews, we identified discrepancies between the understanding of the interviewee and the goals of the survey questions by comparing the interview findings against findings from related work.

3.2. Survey Participants

The survey questions are shown in Table 1. There are 17 questions, 14 closed and three open-ended. We targeted developers who are currently working in the software industry of Bangladesh. We applied purposive sampling [17] to include respondents in a software development related role. Purposive sampling is basically based on the assumption of the population. It is possible that some elements will not have a chance of selection in this method. Moreover, the probability of selection can't be accurately determined in this process. We shared the survey link through the authors' personal connection and in the

Table 1: Survey questions (without demographic) along the four dimensions (D). Subscripts with a question number show # of responses and question type (O = Open, C = Closed)

D1	**Software development methodologies used**
$6_{127,C}$	Which of the following software development methodologies do you follow?
$7_{120,C}$	Which of the followings do you use for requirements gathering?
$8_{126,C}$	On which software development activities, do you spend most of the time?
D2	**Software tools and techniques used**
$9_{128,C}$	Which of the following technologies do you have experience working in?
$10_{127,C}$	What is the primary operating system you are developing on?
$11_{128,C}$	Which programming languages are you using?
$12_{109,C}$	Which frameworks are you using?
$13_{125,C}$	Which IDE are you using?
D3	**Software testing and devops practices used**
$14_{117,C}$	What types of software testing practices do you use?
$15_{118,C}$	What is the level of automated testing in your projects?
$16_{83,C}$	Which tools do you use for testing and quality assurance?
$17_{60,C}$	Which tools do you use for continuous deployment?
$18_{121,C}$	Which version control tool do you use?
D4	**Security and performance measures used**
$21_{74,O}$	How do you ensure scalability of your products?
$22_{73,O}$	How do you maintain performance of your products?
$23_{74,O}$	How do you ensure security of your products?

local developers' groups on social media to achieve our sampling goal. We also implemented the chain referral strategy [18] and asked others to pass on the survey invite. Due to such snowball approach to recruit survey participants, it is not possible to calculate the response rate of our survey. We have conducted the survey through Google Forms. The survey link was opened before the invitations were sent, and the survey link was closed for two consecutive weeks without any response. The survey link was open for feedback for about two months. In total, we received 137 responses from the survey. Each participant was first asked a series of demographic questions (e.g., roles, experience, gender) and then was presented the survey questions related to the development practices. Table 2 shows the distribution of the survey participants by their roles. We noticed that a significant number (69%) of our respondents are developers. Since software developers/developers is a generic role, this can be noticed in other surveys on the SE industry, like the Stack Overflow survey[19, 20]. Previously conducted studies on the Canadian[1] and Turkish[2] SE industry found that more than 80% of respondents were developers. Other roles for respondents to our survey are managers(16.9%), and other kinds of software engineers (8%) (e.g., data engineer, R&D engineer).

Table 3 shows the distribution of the survey participants by their experience. More than 61% of the participants worked in the industry for at least 2 years. In terms of work experience, the demographics of our survey are similar to previous surveys. 77% of our respondents have less than ten years of experience. In the Canadian and Turkish SE industry survey, the percentage of respondents with less than ten years of experience is 67.9% and 79%, respectively. About 38.5% of the respondents in our survey came from three software companies, while the rest (61.5%) were from 38 different companies. About half of the companies (51.9%) are involved in web development, application development, and ERP software development. However, the companies offer a variety of products such as IoT-based health monitoring, cloud services, telecom, ride-sharing platform, biometrics-based personal identity management, and security solutions. According to BASIS, about 1100 software farms are employing 300,000

Table 2: Role wise Distribution of Participants

Role of the Participants	**Percentage**
Developer	69.72%
Manager	16.9%
SQA Engineer	7.04%
Business Analyst	1.4%
R&D Engineer	1.4%
Data Engineer	0.7%
Software Architect	0.7%
Team Lead	0.7%
Trainer	0.7%
UX Designer	0.7%

Table 3: Experience wise Distribution of Participants

Role of the Participants	**Percentage**
less than 2 years	33.58%
2 to 5 years	24.82%
5 to 10 years	18.98%
more than 10 years	17.52%
experience not disclosed	5.11%

IT professionals. BASIS has listed [21] the top ten firms in terms of revenue tax. Around 40% of our respondents are from these software firms and the rest 60% of the respondents are from the 23 other software firms focusing on all the major sources of software products that are being developed in Bangladesh (e.g., mobile, web, B2C, B2B, etc.).

3.3. Data Analysis to Answer RQ1 and RQ2

The survey consisted of both closed and open-ended questions. We analyzed the closed questions using standard descriptive and statistical techniques. We analyzed the closed questions following principles of open coding. Open coding includes labelling of concepts/ categories in textual contents based on the properties and dimensions of the development entities (e.g., tools, processes, etc.) about which the contents are provided. A systematic qualitative data analysis process was followed to analyze the open-ended questions. First, the two authors independently coded the first 30% responses of each question's to extract potential categories. Second, the authors conducted discussion sessions to develop a unified common coding scheme for each question using these categories. Third, the rest of the responses were coded using this coding scheme using the Coding Analysis Toolkit (CAT)[22] software. To measure the level of agreement between two coders, we used the online tool Recal2[23] and CAT[22]. The Recal2 calculator reports the agreement using four measures: 1) Percent agreement, 2) Cohen κ[24], 3) Scotts π[25], and 4) Krippendorffs α[26]. It is believed that Krippendorffs α is more sensitive to bias introduced by a coder and is recommended[27] over Cohen's κ[24]. The level of agreement is presented in Table 4. The average κ value was 0.71. It is a common practice that Cohen's κ value between 0.61 and 0.80[28] is considered a 'substantial agreement. In the coding process, a large number of codes were generated from each of the open-ended questions. To help with our analysis, we conducted discussion sessions to identify the codes that express similar themes. After reaching a consensus, we grouped those codes into a smaller number of high-level categories. We have used the statsmodels[29] and scipy[30] modules in Python for statistical analysis.

4. Study Results

We answer two research questions by analyzing of our survey responses and by comparing our results against surveys conducted for other countries:

RQ1. What are the software development practices in an emerging country like Bangladesh? (Section 4.1)

Table 4: The agreement level in the open coding (#QT denotes # quotes used)

	Q21 (#QT = 74)	**Q22** (#QT = 73)	**Q23** (#QT = 74)
Percent	76.6	72.3	73.7
Cohen κ	0.731	0.688	0.71
Scott π	0.731	0.688	0.71
Krippen α	0.732	0.689	0.711

RQ2. How do the development practices in Bangladesh differ from other developed countries? (Section 4.2)

The first research question focuses on the dimensions of software development practices and processes in Bangladesh. The second research question compares those practices prevailing in Bangladesh with other regions to reveal the similarities and dissimilarities in software development practices and processes.

4.1. Development Practices and Processes in Bangladesh (RQ1)

Following previous studies in region-based software development process analysis [1, 2, 3, 4], we analyze the development practices and processes in Bangladesh along four dimensions (D):

D1. Software development methodologies used (Section 4.1.1). We investigate the development processes, methodologies, requirements analysis processes.

D2. Software tools and techniques used (Section 4.1.2). We determine the trending technologies like platforms, programming languages, and frameworks.

D3. Software testing and devops practices used (Section 4.1.3). We explore the present situation of testing and deployment practices.

D4. Performance and security measures used (Section 4.1.4). We analyze how software companies secure and maintain their code and what practices are followed to ensure performance and scalability.

4.1.1. Software development methodologies used (D1)

Software development methodology implies the process used for developing particular software in a structured and methodical way. We asked three questions: 1. Software development methodologies (Q6), 2. Requirements gathering (Q7), and 3. Most time-consuming software development activities (Q8).

- **Software development methodologies (Q6).** Agile is the most popular (64%) development methodology in Bangladesh, followed by Scrum (46%). These outcomes match the 2018 Stack Overflow survey [31], which also reports that Agile and Scrum are the most popular methodologies worldwide. Among the other methodologies, pair programming (20%) and Waterfall (12%) are also popular in Bangladesh software companies.

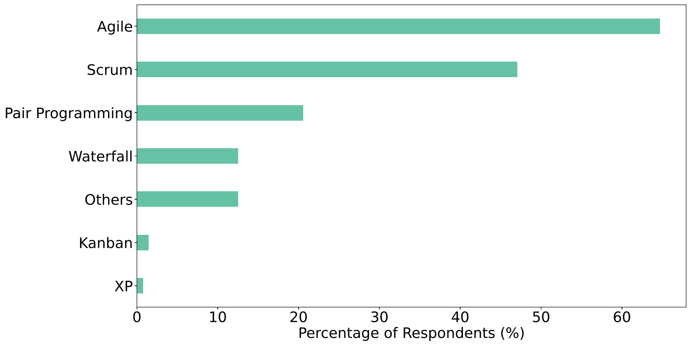

Figure 1: Software development methodologies (Q6)

- **Requirements gathering (Q7).** The most critical activity that always arises during software development is collecting and analyzing the requirements of a system. Usually, the outcome of the analysis is presented before the client and modified according to their feedback. The more clear and detailed requirements are, the higher the possibility of building software that conducts the clients anticipation. Corresponding to Figure 2, using plain text (44%) and storyboard (41%) are the most widely used requirement gathering techniques among our survey participants. The other relevant techniques include use case (36%), GUI prototype (35%), grooming session (30%), etc.

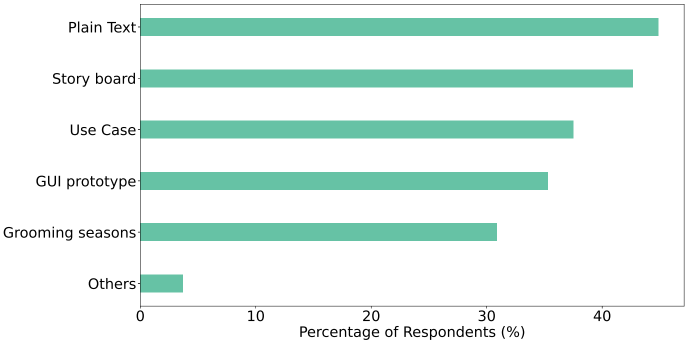

Figure 2: Requirements gathering (Q7)

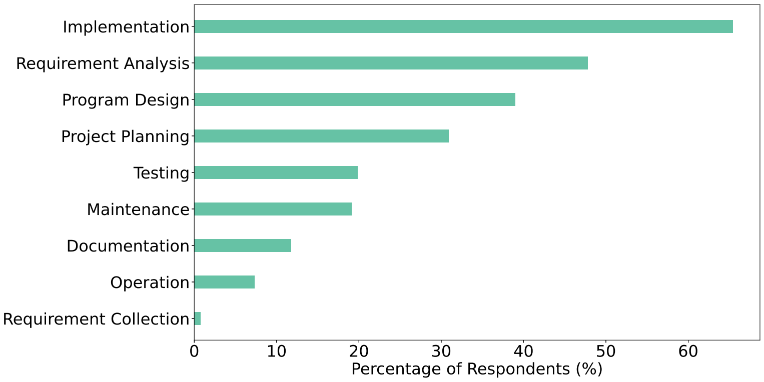

Figure 3: Timeline of Development Activities (Q8)

- **Timeline of Development Activities (Q8).** According to 65% of our respondents, most of the time is spent in the implementation stage, whereas the requirement analysis stage requires the second most according to 45% response. The other usages are program design (37%), project planning (30%), testing (19%), maintenance (17%), etc. There is no significant correlation between software development methodologies (Q6) and the most time-consuming development activities (Q8) based on Cramér's V [32] analysis. Cramér's V calculates the association between two nominal variables using Pearson's chi-squared statistic [33].

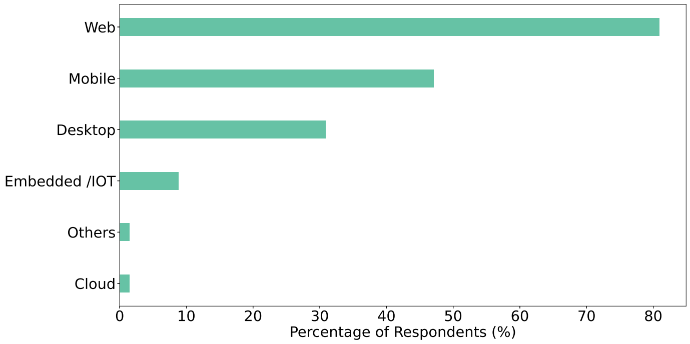

Figure 4: Technology Platforms (Q9)

RQ1-D1. Software development methodologies used. To provide software services, companies prefer the Agile approach most followed by Scrum and also collect requirements via plain text in a high percentage. Besides, developers of the Bangladeshi SE industry generally spend more time on implementation-related activities than planning and testing.

4.1.2. Software development tools and techniques used (D2)

We ask five questions related to the adoption of technologies and tools by the participants: 1. Technology Platform (Q9). 2. Operating System (Q10). 3. Programming Language (Q11). 4. Framework (Q12). 5. IDE (Q13).

- **Technology platforms (Q9).** 80% of survey respondents work for the web platform (Figure 4). The rests are mobile (45%), desktop (30%), embedded/IoT (8%). This distribution is similar to the 2020 wordwide survey of JetBrains [34], which finds that websites are the most common type of application developers work on, and the web platform is the most preferable and popular to develop, followed by desktop and mobile. We have conducted a cross-aspect analysis to identify any relationship between the technology platform and the requirement gathering process. The bubble charts in Figure 5 visualize the cross-aspect analysis. It is clear from the figure that the requirement gathering process is mostly practiced in GUI-based development (e.g., web, desktop, mobile).

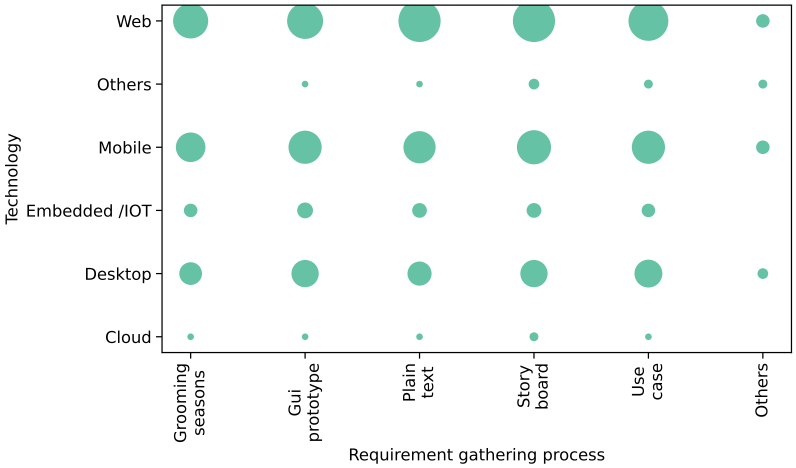

Figure 5: Cross aspect analysis of requirement gathering and technology platform

- **Operating Systems (Q10).** Most of our respondents preferred Linux based operating system (56%) for their development (Figure 6). The second frequently used operating system is Windows (45%) followed by MacOS (28%). We observed similar scenarios in the 2018 and 2019 StackOverflow (SO) surveys [31, 20]. However, Windows ranked first in the 2020 survey of both SO and JetBrains [19, 34]. The recent higher preference towards Windows could be due to the newly included WSL (Windows Subsystem for Linux). This feature allows users to perform almost any Linux-specific task on Windows. We anticipated that the use of OS might be related to professional experience. Among the participants, senior/expert developers (those with at least 5 years of experience) have significantly higher rates of Linux usage ($p = 0.024$ based on Mann Whitney U test).

- **Programming Languages (Q11).** Around 65% and 60% of our respondents use JavaScript and Java, respectively, which are the two most used languages in Bangladesh (Figure 7). Both JavaScript and Java are popular for web and mobile platforms. A great percentage of our survey participants develop for both web and mobile platforms. Other languages like PHP (25%), Python (25%), and C# (18%) are also used, which indicates that the software engineers are not bi-

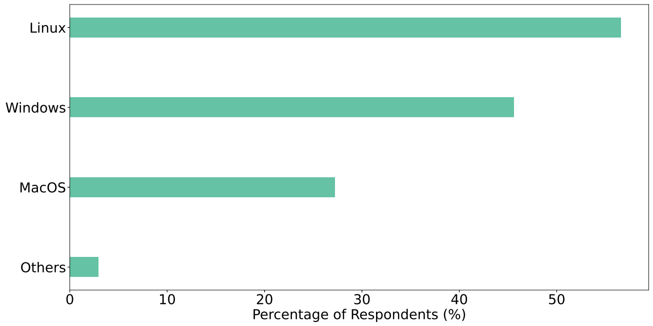

Figure 6: Operating Systems (Q10)

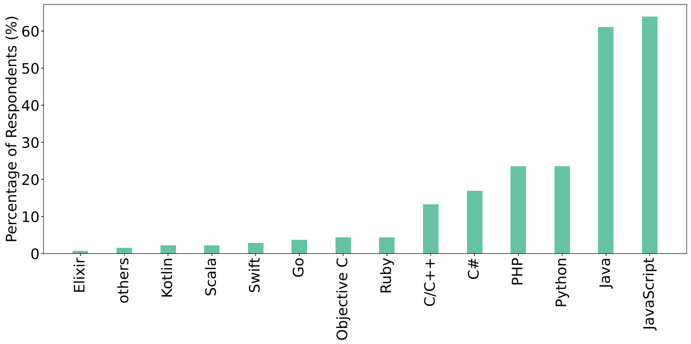

Figure 7: Languages used in software development (Q11)

ased towards a single specific language. Our survey result matches with the last two years' Stack Overflow survey and the GitHub stat. In all of the cases, JavaScript is the most used language, followed by Java and Python [19, 20, 35]. We observed that users using mobile and web platforms mostly use Java and Javascript as programming languages. However, this is not statistically significant ($p = 0.1$). Though the use of the operating system can be influenced by programming language (e.g., Swift and macOS), we do not find any significant correlation between the two choices.

• **Frameworks used in development (Q12).** As shown in Figure 8 Spring boot (37%) is the most used in the Bangladesh software industry. This observation is aligned with the result of Java's usage rate corresponding to Figure 7. Since JavaScript is the most used language of our respondents, they use

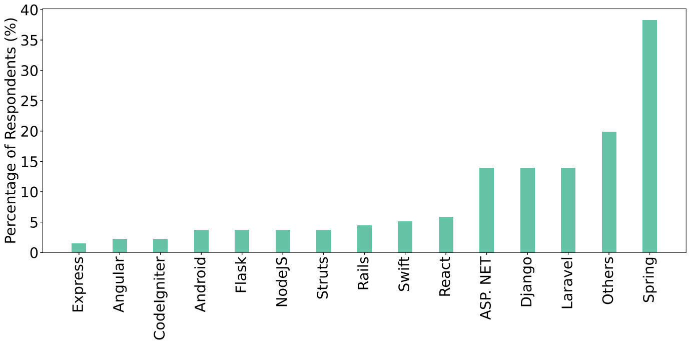

Figure 8: Frameworks used in development (Q12)

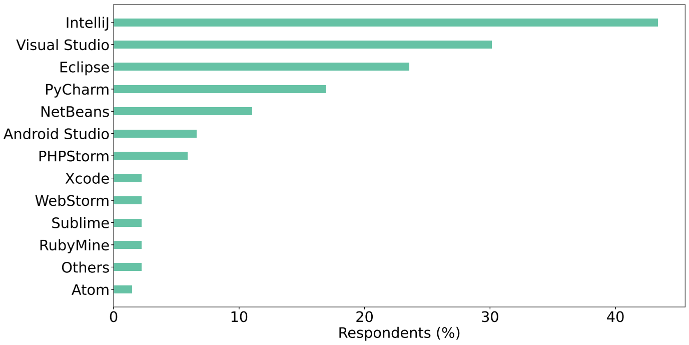

Figure 9: IDEs used by the respondents (Q13)

various JavaScript frameworks such as React, Node.js, Angular, Express, etc. ASP.NET, Django, and Laravel are used in the same proportion based on around 15% of our respondents. React, Swift, Ruby on Rails, Node.js, etc., are comparatively less used. Other than these, lots of frameworks such as Cocoa, Meteor, TestNG, Relay, Appium, CakePHP, etc., are also used (presented as 'Others' in Figure 8). For web development, Django and Spring frameworks are mostly used in Bangladesh ($p = 0.04$). We have compared our results with the Stack Overflow 2016 to 2020 survey[36, 31, 20, 19]. The only common framework in the top five list in both surveys is ASP.NET. In the stack overflow survey, we noticed that JavaScript-based frameworks (e.g., Jacqueline, Angular, React, Node.js) occupy the top positions (top five), which is not the case for our survey.

• **IDEs used by the respondents (Q13).** As shown in Figure 9, IntelliJ, is used by most respondents (43%). IntelliJ is a Java integrated development tool for developing software for the enterprise, mobile, and web application, The other IDEs used in SE industries are Visual Studio (30%), Eclipse (24%), PyCharm (17%), NetBeans (11%), and Android Studio (7%).

RQ1-D2. Software development tools and techniques used.
Web-based software services top the list of development technologies. The requirement gathering process is mostly practiced in GUI-based development. Practitioners prefer Linux-based operating systems (OS) most though other OSs (e.g., Windows, macOS) also have an appreciable usage rate. JavaScript and Java are the two most popular languages. Hence, Java integrated tool, IntelliJ, tops the list of IDEs followed by Visual Studio and Eclipse.

4.1.3. Software testing and devops practices used (D3)

We asked five questions to determine the adoption of testing and devops techniques in the Bangladesh software companies: 1. Software Testing Practices (Q14), 2. Level of Automated Testing (Q15), 3. Tools Used in Testing and QA (Q16), 4. Continuous Deployment tools (Q17), and 5. Version Control (Q18).

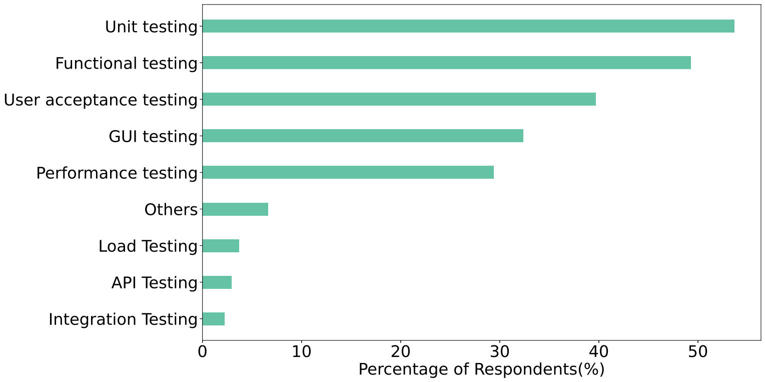

Figure 10: Software Testing Practices (Q14)

• **Software Testing Practices (Q14)**. According to Figure 10, several testing practices are used during software development. The results show that most of the organizations have carried out unit testing (53%), functional testing (49%), user acceptance testing (39%), GUI testing (31%), etc. Unit testing also ranked first in the 2019 survey of JetBrains, where it was voted by 71% participants across the globe [37]. We observed in our survey that in some cases managers have reported performing GUI testing and performance testing, which is unlikely in their role/designation. However, the observation is not statistically significant ($p = 0.12$). It may be deduced that in absence of enough specialized resource managers have to take additional responsibility. To identify the relation between testing practices and experience, we plotted them together in Figure 11. We have observed that junior developers (less than 5 years of experience) mostly perform unit, integration, and functional testing, whereas senior developers mostly perform API testing. We conducted the Mann Whitney U test to assess the conjecture, and it is found statistically significant ($p < 0.01$).

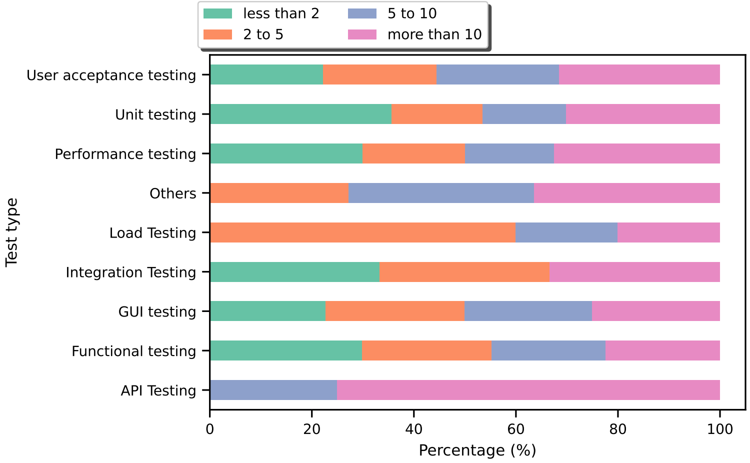

Figure 11: Testing practices ans professional experience

- **Level of Automated Testing (Q15).** We have asked the survey participants about the level of automated testing performed in their respective companies. The responses were gathered using the Likert scale. It was found that different respondents have very different experiences in this context, i.e., some companies heavily practice automated testing, while others favor manual

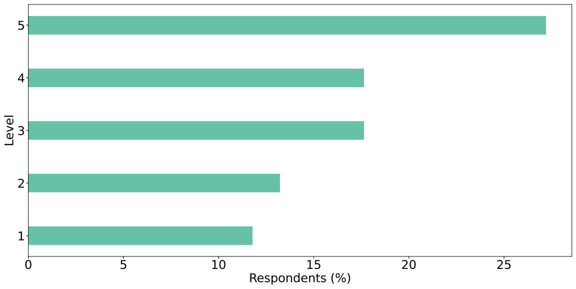

Figure 12: Level of Automated Testing (Q15)

testing. Results are shown in Figure 12, which indicates that about 70% of our respondents (others than who voted for level 5) do not use automated testing regularly. The level of automated testing might be related to the programming language/framework. The testing suite provided by framework/language might encourage developers to implement automated testing. The level of automated testing vs. language and framework is plotted in Figures 13 and 14, respectively. It seems from Figure 13 that the highest level of automated testing is mostly practiced in Java, JavaScript, Objective-C, and PHP language. We conducted the Mann Whitney U test to assess our conjecture and it is found statistically significant ($p = 0.01$). From Figure 14, we found that the highest level of automated testing is mostly performed in Android, Express, Node.js, Struts, and Java EE framework, and the observation is statistically significant ($p = 0.006$). Also, the highest level of automated testing is mainly used by developers (unit testing). Managers practice the lowest level of automated testing. We observed that managers mainly engaged in assessing the acceptability of the product from the end-user point of view. We plotted developer experience vs automated test levels in Figure 15. Junior developers tend to use more automated testing than senior developers. However, the observation is not statistically significant ($p = 0.08$). One reason behind this observation may be that the senior developers perform GUI testing more, which are hard to automate.

• **Tools Used in Testing and QA (Q16).** Most of the respondents have used XUnit (e.g., JUnit, NUnit) (30%), Selenium (27%), Jenkins (20%), others (9%)

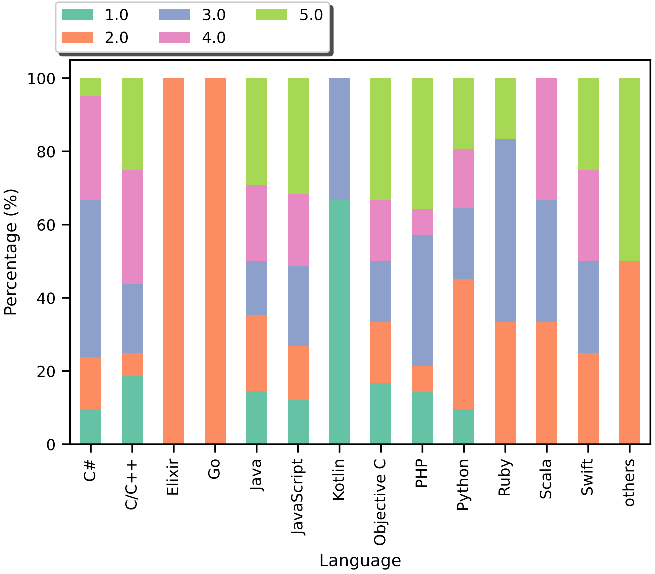

Figure 13: Programming language and automated testing level

(see Figure 16). Around 38% of our respondents were not interested in replying to this question that is not surprising because the majority of the respondents (93% approx.) were working on roles other than SQA engineer as per Table 2, and they are not supposed to be involved in any testing themselves.

- **Continuous Deployment tools (Q17).** majority of the respondents deploy their implemented codes using AWS code-deploy (12%) and Jenkins (12%) (Figure 17). The other deployment tools are Bamboo (5%), TeamCity (4%), Octopus (2%), etc. Respondents voted none (4%) as they didn't use any deployment tools and 53% of the respondents were not interested in this topic. Anyhow, the percentage of uninterested respondents does not seem unexpected. From Table 2 and 3, we can observe that a significant portion of our respondents is developers, and more than half of our respondents are experienced for less than five years respectively. As deployment is related to DevOps, it is quite likely that developers have not enough knowledge or have less interest in deployment. The outcome indicates that the usage rate of deployment tools in Bangladesh

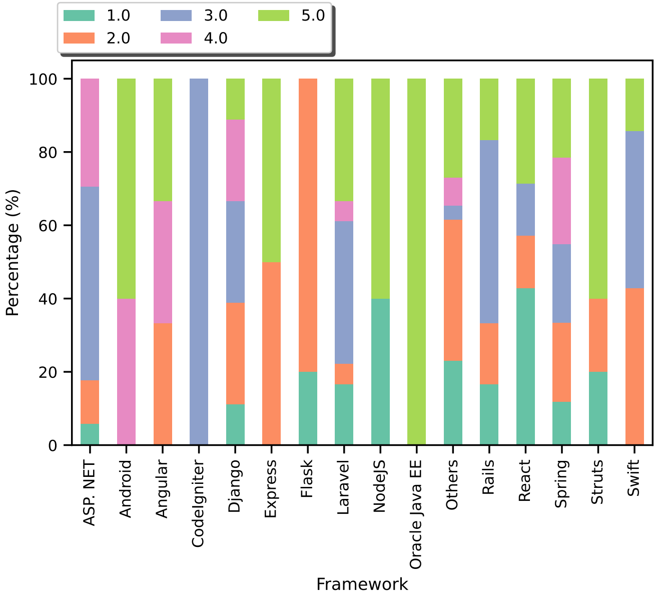

Figure 14: Framework and automated testing level

for continuous integration and continuous deployment is not widespread yet.

- **Version Control (Q18).** Git (78%) and Bitbucket (29%) are mostly-used version control systems in the software industry as shown in Figure 18. Besides these, Subversion (SVN) (5%) and others (4%) are used. Respondents were allowed to select more than one option. The 2018 Stack overflow survey[31] reports that the most popular version control system is Git (87.2% developer uses Git) and the second most popular is SVN (16.1% developer uses SVN). However, in our survey, we found a slightly different result, the most popular version control system is Git and the second most popular is Bitbucket. This might be related to the declining popularity of SVN over the years. From the Stack overflow survey over the range 2017-2018, it is clear that SVN is losing popularity to Git.

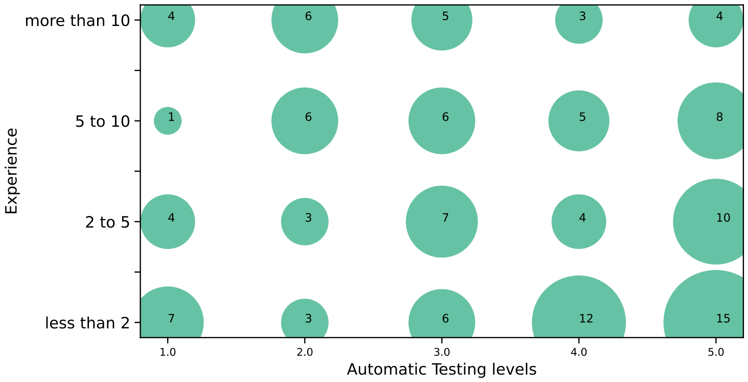

Figure 15: Experience and automated testing level

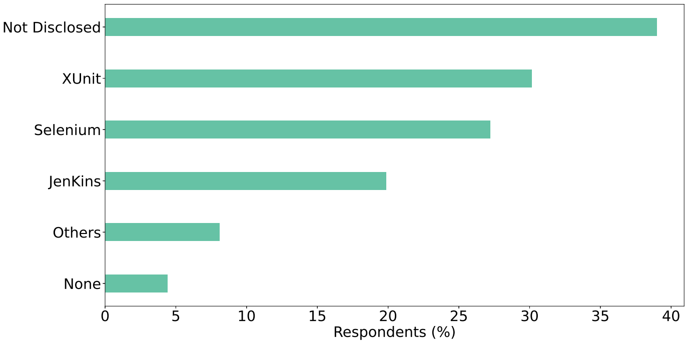

Figure 16: Tools Used in Testing and QA (Q16)

RQ1-D3. Software testing and devops practices used. Unit testing is heavily carried out by developers in the Bangladesh SE industry likewise across the globe. In addition, various software testing tools (e.g., XUnit, Selenium, etc.) and deployment tools (e.g., AWS code-deploy, Jenkins, etc.) are used. However, there is a tendency among most Bangladeshi developers not to use these automated tools regularly. Git is the most popular version control system, while SVN is losing its popularity in the Bangladesh SE industry likewise across the globe.

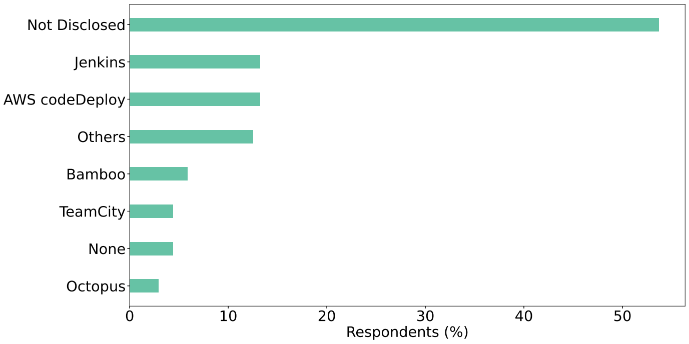

Figure 17: Continuous Deployment tools (Q17)

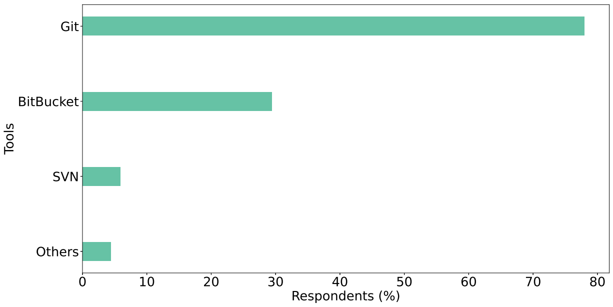

Figure 18: Version Control (Q18)

4.1.4. Performance and security measures used (D4)

Security and performance are two of the most important non-functional requirements for any software product. We asked three open-ended questions with regards to the enforcement of security and performance-related features in software products in Bangladesh: How do you ensure performance, scalability (Q21, Q22) and security (Q23) in your software products?

- **Performance (Q21, Q22).** Software performance indicates how efficient the software is in terms of response time and resource consumption. We find nine types of performance measures that are practised in software products developed in Bangladesh (see Figure 19). The twelve types are divided into four categories: Use of 1. tools and frameworks (46.72%), 2. design principles/best practices (26.17%), 3. testing (16.82%), and 4. review and feedback (9.35%).

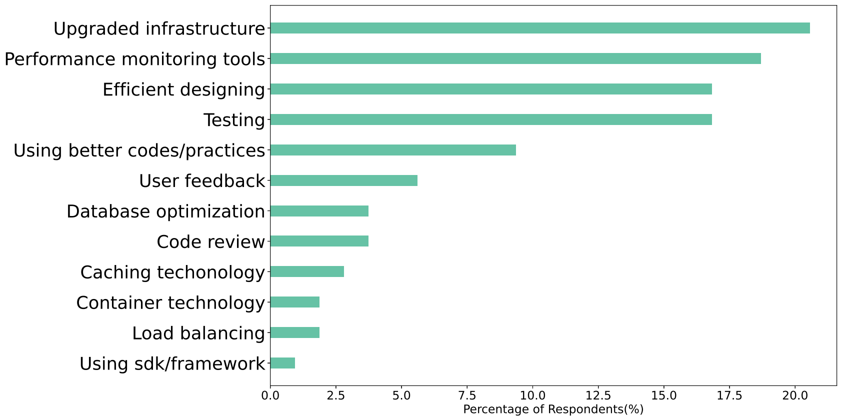

Figure 19: Measures to ensure performance & scalability of products

5. database optimization (3.74%)

(1) ***Exploitation of tools and frameworks.*** Six out of twelve types belong to this category. (a) *Tools*: Around 18.69% respondents use tools and metrics to measure performance: *"take help of different performance monitoring tools and dashboard, analyzed data, measure time and memory efficiency of process"*(S_{35}) (b) *Infrastructure*: Around 20.56% of respondents use upgraded infrastructure to ensure performance like cloud hosting (e.g., Amazon AWS), a high-end server, and new technologies. (c) *Caching*: Around 2.8% respondents implemented caching to maintain software performance. (d) *Container Technology*: Containers enable users to scale their system without any dependency on the underlying OS. About 1.87% of our respondents use container technologies to ensure their products scalability and performance. (e) *Using SDK/framework*: About 0.93% of respondents depend on the framework to maintain software performance and scalability. (f) *Load Balancing*: Around 1.87% respondents use load balancing as a measure to maintain performance: *"Optimizing number of HTTP requests, Asynchronous programming, Caching, CDN, Load Balancing, nginx, varnish, compression of data, Continuous monitoring, Load testing, stress testing"*(S_{42})

(2) ***Use of design principles/best practices.*** Around 26.17% respon-

dents try to ensure software performance right from the design phase as follows. (a) *Using better codes/practices*: Around 9.35% respondents ensure performance by implementing industry-standard best practices like compression technology, enforcing design patterns, and refactoring. (b) *Efficient designing*: Around 16.82% of respondents emphasize on performance-aware architecture design.

(3) ***Use of testing.*** Around 16.82% respondents rely on the software testing strategy to ensure performance like load testing and stress testing.

(4) ***Use of review and feedback.*** Around 9.35% of our respondents use user feedback (e.g., continuous feedback from QA team S_{65}) and code review to improve product performance. According to S_{15}: *"The code quality is assessed by the different team members during code review, followed by designing new ways to solve issues in the product that are time-intensive."*

(5) ***Database Optimization.*** Around 3.74% of our respondents use database optimization to ensure performance and scalability. Database optimization includes sharding, clustering, indexing, and scaling. According to S_{85}: *"Besides scaling horizontally, database scaling is performed by partitioning tables, along with multi-threaded implementations"*

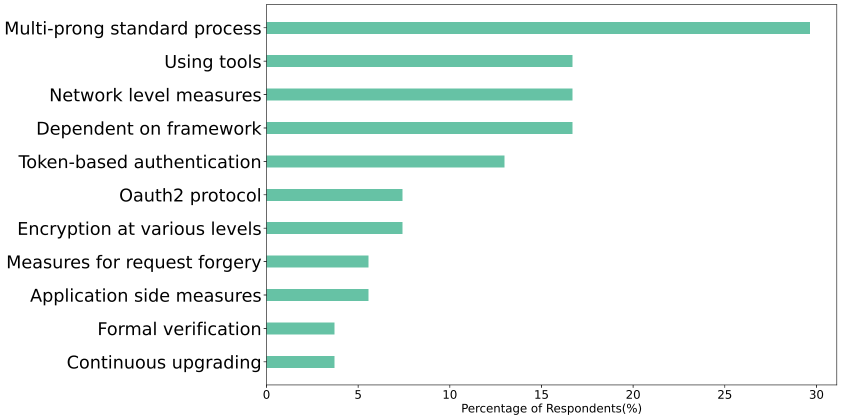

Figure 20: Measures to ensure security of products

• **Security (Q23):** Our open coding of the survey responses reveals 11 labels (see Figure 20). The 11 labels are are divided into three main categories of security-related development practices: 1. Measures related to authentication and authorization (64.82%), 2. Exploitation of tools and techniques to ensure security in products (53.71%), and 3. Use of encryption technologies for data (7.41%). We discuss the categories below.

(1) ***Measures related to authentication and authorization.*** Six out of the 11 labels belong to this category. (a) *Multi-prong Standard Process.* About 29.6% of the respondents reported that they practice various security standards and protocol to ensure security (e.g., ISO/IEC 27001, PA DSS). (b) *Token-based authentication.* About 13% respondents reported to have implemented a token-based authentication system, which allows users to enter their username and password to obtain a token for authentication and authorization. (c) *OAuth 2.0* : Around 7.4%) respondents use the OAuth 2.0 protocol as the primary way of maintaining security. OAuth 2.0 is the industry-standard protocol for authorization. OAuth 2.0 focuses on client developer simplicity while providing specific authorization flows for web applications, desktop applications, mobile phones, and living room devices. (d) *Application-side measures*: Around 5.6% of respondents implement security measures at the application level like encryption of application data at the client-side, use of https while pulling data from a server, secured architecture, etc. (e) *Measures for request forgery*: Around 5.6% respondents implemented security measures against Cross-site request forgery (e.g., attacks like cross-origin resource sharing (CORS), cross-site request forgery (CSRF) or one-click attack or XSRF). Security testing is paramount for this: *"Security testings like: SQL injection, cross-site scripting, CSRF, API security, use of https, detecting malicious/suspicious HTTP requests and auto-blocking"* (Respondent ID S_{42}) (f) *Formal Verification*: Around 3.7% respondents ensured the practice of formal code review to enforce security practices: *"There are some basic guidelines that we must follow and while code review this needs to be an absolute part that needs to be checked before the code gets merged"* (S_{112})

(2) ***Exploitation of tools and techniques to ensure security in products.*** Four out of the 11 labels belong to this category. (a) *Dependent on Framework*: Around 16.7% respondents depend on the underlying framework

for security like Spring, HDIV, and Laravel. *"https, popular framework which already prevents some type of attacks. rest of the things on case-by-case basis)"*(S_{79}) (b) *Use of tools*: Respondents use various open-source/paid tools for scanning and testing like OWASP and penetration testing tools. (c) *Network level Measures*: Network-level measures include IP-white-listing, port-blocking, VPN, and the use of HTTPS in software. 16.7% of respondents use at least one of the mentioned strategies to ensure security. (d) *Continuous Upgrade*: Around 1.5% respondents reported that they arrange frequent hackathons, workshops, and security audits: *"We run security audit of our office environment. We also conduct security session per 6 months to introduce latest trend in threats and what we can do to avoid it"*(S_{57})

(3) ***Use of encryption technologies for data***. Around 7.4% respondents use encryption at the different levels of software architecture such as network, data, and transmission. *"use encryption at different level of software (server, network, transmission layer, database and software layer.)"*(S_{35})

RQ1-D4. Security and performance measures used. Bangladesh software industry uses various methods to ensure security, but the adoption of tools is not widespread. We have noticed that the Bangladesh software industry mostly uses performance monitoring tools and software testing to ensure product performance. Software scalability is generally considered at the design stage (e.g., efficient design) in the Bangladesh SE industry.

4.2. Comparison of Development Practices Among Countries (RQ2)

Given Bangladesh is a developing country with a rapidly emerging software development industry, our observations of Bangladesh SE industry can offer insights into how such an industry is operating in other similar countries and how such observations could differ from countries with matured SE industries. In this section, we offer a comparative assessment of the our observations in Bangladesh SE industry against observations reported previously in other countries like Canada, Turkey, Netherlands, New Zealand, etc. [1, 2, 3, 4]. Similar to RQ1, we present the comparisons along four dimensions: D1. Software development methodologies used (Section 4.2.1), D2. Software tools and techniques used (Section 4.2.2), D3. Software testing and devops practices used (Section 4.2.3), and D4. Security and performance measures used (Section 4.2.4).

4.2.1. Software development methodologies used (D1)

We compare the observations from three questions in our survey: 1. Software development methodologies (Q6), 2. Requirements gathering (Q7), and 3. Most time-consuming software development activities (Q8). We summarize the comparisons in Table 5 and discuss those below.

• **Software development methodologies (Q6).** Our study shows that the most acceptable method in Bangladesh is the agile model (64%) likewise across the globe [31]. However, the usage of the scrum (44%) in New Zealand has better usage followed by agile (30%) [4], and in Turkey, the waterfall model is mostly used based on the survey of Garousi et al. [2]. Again, in both Bangladesh and New Zealand, extreme programming (XP) has a lower percentage of usage. Almomani et al. [38] found that software developments in Malaysia are predominantly regulated through ad-hoc approach (53%) and the agile methodologies (46%) since usually software organizations are majorly concerned with short-term delivery of software products.

• **Requirements gathering (Q7).** According to the Figure 2, using plain text (44%) and storyboard (41%) are the most widely used requirement gathering methods. This result is similar to the survey of Vonken et al. [3]. From their study, we can find that the textual description of specifying requirements is the most favorite approach in the Netherlands.

• **Timeline of Development Activities (Q8).** According to the study of Wang et al. [4], during system design and development, most time is spent on implementation and coding, and relatively less time is spent on maintenance in New Zealand similar to Bangladesh, as revealed in our study. On the other hand, requirement analysis requires the second most time in Bangladesh, ac-

Table 5: Comparison of development methodologies, requirements collection and activities between our findings and prior findings

Practices	Our Study	Prior Study	Comparison
What software methodology are used in your project?	(1) Agile and scrum are the two most used methods for development. (2) Appreciable respondents answered that plain text and storyboard for requirements gathering, which shows that tools for requirements collection yet are not used usually in Bangladesh. (3) The implementation stage gets more preference than system design, planning, and testing in Bangladesh.	The usage of the scrum and waterfall is high in New Zealand [4] and Turkey [2] respectively. Software developments in Malaysia are predominantly regulated through ad-hoc approach, and the agile methodologies [38]. The textual description of specifying requirements is a firm favourite in Netherlands [3]. Specified design documentations are well-practiced rather than just implementation without well-defined planning and documentation in India, Japan, and Europe [7]. However, most time is spent on implementation and coding in New Zealand [4].	We have perceived that in the Bangladesh SE industry, there exists an ad-hoc based software development practice which might be similar to the practices in Malaysia nonetheless, the regions like Japan, Europe, etc., give more importance to system planning and design that makes those region-specific SE industries more stable.

cording to 45% respondents of our survey. In contrast, in Malaysia, as per [5], most organizations spend from 5% to 20% of their efforts on planning and requirement analysis. However, if we compare Bangladesh with technologically advanced regions like Japan, India, Europe, etc., with the help of the study of Cusumano et al. [7], we observe that there exist a substantial difference in the timeline of development activities with Bangladesh. Their study has reported that architectural, functional, and design specification documents are the most used and well-regarded practice in those regions rather than just writing code with minimal planning and documentation. But in Bangladesh, the implementation phase gets the most priority over other development stages.

RQ2-D1. Software development methodologies used. The agile method is the most popular approach for software development across the globe except in some areas (e.g., New Zealand). Bangladesh SE industry mainly uses plain text like other countries to collect requirements (e.g., Netherlands). In comparison with technologically advanced regions, Bangladesh SE industry lags in giving value to system design and planning.

4.2.2. Software development tools and techniques used (D2)

We compare our observations from three questions: 1. Technology Platform (Q9). 2. Operating System (Q10). 3. Programming Language (Q11). The observations from two questions (Programming language Q11 and IDE used Q12) could not be compared, because those were not previously asked in the context of other countries. We summarize the comparisons in Table 6 and discuss those below.

• **Technology platforms (Q9).** As shown in Figure 4, most of our survey respondents (80%) work in web platforms. This outcome is similar to the result of the survey of Wang et al. [4] in which the authors found that most of their respondents also develop in web platforms in New Zealand.

• **Operating Systems (Q10).** We have found an interesting point that Windows is mostly used among developers of New Zealand based on the study [4]

Table 6: Comparison of tech platforms, OS and programming languages between our findings and prior findings

Practices	Our Study	Prior Study	Comparison
Which implementation technologies and tools are adopted by software development professionals?	(1) Majority of our participants were related to the various types of web-based applications that convey the market demand for web-based apps in Bangladesh. (2) Linux is mainly used for development purposes while macOS is highly preferred by managers. (3) Javascript and Java are two mostly used programming languages in the Bangladesh SE industry since the demand for web-based services as well as mobile apps.	There exists a high level focus on web-based platforms in the software industry of New Zealand [4].	We have observed that web-based platforms have widespread demand among different countries along with Bangladesh. Besides, in using programming languages like Java and python, Bangladesh has similarities with Turkey but is different from New Zealand.
		Windows is highly preferable among developers of New Zealand [4].	
		The use of Java as a programming language is spacious in Turkey [2]. However, in New Zealand, the usage rate of Java, as well as python ranks somewhat low [4].	

nevertheless, Linux is mostly used in the case for Bangladeshi developers corresponding to our survey presented in Figure 6.

• **Programming Languages (Q11).** According to the study of Wang et al.[4], Java ranks quite low in New Zealand; nevertheless, it is the second most used programming language in Bangladesh as per our study reported in Figure 7 as well as the most used language in Turkey [2]. Again, Python did not have a good standing in the ranking of languages used in New Zealand; nevertheless, it is used significantly in Bangladesh.

RQ2-D2. Software development tools and techniques used. Like other SE industries, the web is the main technology platform in Bangladesh. Linux is the preferred OS in Bangladesh's SE industry, while it is Windows in New Zealand SE industry. Although Java and Python are popular languages in the SE industry in Bangladesh, we have noticed that they are not very popular in the New Zealand SE industry.

4.2.3. Software testing and devops practices used (D3)

We compare the observations from three questions in our survey: 1. Software Testing Practices (Q14), 2. Level of Automated Testing (Q15), The observations from three questions (Tools Used in Testing and QA Q16, Continuous Deployment tools Q17, and Version Control Q18) could not be compared, because those were not previously asked in the context of other countries. We summarize the comparisons in Table 7 and discuss those below.

• **Software Testing Practices (Q14)**. From our study, as per 10, we see an interesting point that unit testing (53%) and functional testing (49%) are moderately used in Bangladesh, whereas from [1] and [4] we can see that relatively a high percentage of their survey respondents in both Canada New Zealand rely on unit testing with 79.27% and 73% respectively. On the other hand, the adoption of acceptance testing and UI testing is quite similar to these countries. In Malaysia, based on [5], Baharom et al. reported that, according to their survey, unit testing (68.29%), integration testing (78.05%), system testing

Table 7: Comparison of testing practices and test automation between our findings and prior findings

Practices	Our Study	Prior Study	Comparison
What type of testing and deployment practices are used?	(1) Around half of our respondents responded about carrying out unit testing and functional testing. Also, the utilization rate of acceptance and UI testing has an appreciable percentage in Bangladesh. (2) Automated testing exercise is not usual in Bangladesh as per the majority of our respondents.	Unit testing is observed to be the most exercised in Malaysia (68.29%) [5], Canada (79.27%) [1], New Zealand (73%) [4] in a great percentage. Test automation is highly embraced by France software industry and comparatively less adopted in overall Europe [39].	We have perceived that unit testing is moderately practiced in Bangladesh, though its usage is comparatively spacious in other countries. In the adoption of test automation, the software industry of Bangladesh is way behind France though the usage rate might have similar to other European countries.

(85.37%), and acceptance testing (78.05%) are used by most organizations in a high percentage, and about half of the organizations are carrying out alpha and beta testing.

- **Level of Automated Testing (Q15).** We have found that as per 12, around 25% of our respondents are highly concerned that they have to use automated testing for their projects, while around 35% of our respondents have expressed medium level concern and the remaining are hardly concerned about using automated testing. From the study of Dutta et al. [39], we have found that in automated testing practices, Bangladesh is quite similar to all of Europe but lags behind France. According to their study, the usage rate of automation testing tools in overall Europe is 26%, where in France, it is as high as 61%. But in Israel, this rate is the only 9%.

RQ2-D3. Software testing and devops practices used. In the adoption of test automation and widespread unit testing, the Bangladesh SE industry lags behind developed countries' SE industry. Software developers around the world usually give unit testing the top priority, but the developers in Bangladesh have comparatively less participation.

4.2.4. Performance and security measures used (D4)

We compare observations from our three open-ended questions with regards to the security and performance-related features: How do you ensure performance, scalability (Q21, Q22) and security (Q23) in your software products? We summarize the comparisons in Table 8 and discuss those below.

- **Performance (Q21, Q22):** 21.82% respondents of our survey use performance testing to ensure the performance of their product. However, it is the second least practiced measure among all the measures. Garousi et al.[2] found that developers mark the lack of performance testing as the main challenge in software maintenance in the Turkish software industry. However, the scenario is different for the Canadian software industry. Participants of the survey of Garousi et al.[1] reported that 40% of them conduct performance testing, and 30% of their total testing effort is spent on performance testing. The New Zealand software industry follows the practice similar to Canada as reported by Phillips et al.[43]. The practice in Bangladesh matches that of Pakistan. In the survey of Shah Jahan et al.[44], only 5% of participants reported conducting performance testing. It seems that performance testing is less popular in

Table 8: Comparison of security, performance, scalability between our findings and prior findings

Practices	Our Study	Prior Study	Comparison
How are the security and performance is ensured in a product of a company?	(1) The practice of security architecture and security testing as a security measure are not prevalent in the Bangladesh SE industry. (2) Performance testing and peer review are the two least practiced performance measure in the Bangladesh SE industry. (3) Cloud service and architecture are two of the top three scalability measures.	Security testing is found to be the least practiced in the software industry in Turkey[2], Malaysia[40], India[41], and New Zealand[42] Performance testing is a common practice in some [1, 2, 43] SE industry. However, it is hardly practiced in the Pakistan[44] SE industry. Also, peer review is a common practice among Turkish[2] developers. To ensure scalability, there are many practices prevalent in other countries' SE industry, such as microservice architecture[45] and containerization technologies[46].	We found that the practice of 'no security' measures is less in Bangladesh compared to other countries. Though the practice of security testing Bangladesh has similarities with other countries, it lags behind the practices of matured industries. However, like the matured and grown software industries, cloud service is a popular measure in Bangladesh.

growing software industries such as Bangladesh and Pakistan.

Peer review is the least practiced measure in the Bangladesh software industry. However, in the Turkish software industry, peer review is ranked as the most frequent activity[2] (ranked five on a five-point Likert scale), though the practice is only limited to code review. Architecture/design review is hardly practiced in turkey (ranked one on a five-point Likert scale). We found that peer review is limited to only code review in the Bangladeshi software industry, and the only 7.27% of our participants reported practicing peer review.

There is no study focusing on scalability practices in specific software industries, so it isn't easy to compare scalability practices. However, in a study on Finnish DevOps, Laihonen[45] found that the Finnish software industry prefers cloud services as it helps them automate quality assurance. He also reported that DevOps is inclined towards micro-service architecture when selecting a product rather than monolithic architecture. Hussain et al.[46] conducted a study to identify trends in the DevOps practices in New Zealand. For this study, besides interviewing the DevOps, they examined the job advertisements for a DevOps role. They found that containerization technologies (e,g., Docker, Kubernetes) have a high demand in the New Zealand software industry. 94% of job advertisement requires expertise in one or multiple containerization technologies. This indicates the popularity of docker technology in the New Zealand software industry. However, in the Bangladeshi software industry, the scenario is different. Cloud services are the second most popular (28.85%) measure to ensure scalability where the use of containerization technologies are not that much popular (3.85%)

- **Security (Q23):** A very few percentages (9.56%) of our survey respondents reported not to use any security measures in their product. This practice is also prevalent in the Indian and Malaysian software industries. Bahl et al.[41] reported that due to misalignment with organization design, goal, and strategy in some Indian software firms, security measures are not practiced. In a study with Malaysian developers, Farvin et al.[40] found that 31% of respondents think it is not required to add security in the requirement analysis of a product. Basharat et al.[47] reported a sense of false security in the small software industry and standard security practices are hardly followed. It is likely to be applicable to the industry in Bangladesh as well. From the response of a survey on the Turkish software industry, Garousi et al.[2] ranked different design activities in terms of frequency. Security architecture was ranked second out of five (five is for always used activities and one is for never used activities). The ranking represents that security architecture is not a frequent activity in the Turkish software industry. However, in our survey, we see very few respondents reported to practice security design principles while designing system architecture. Our survey found that 5.56% of respondents rely on security architecture/security design principles (application side measures) to ensure security in their product. The software industries of Bangladesh, Turkey, and New Zealand have a resemblance in the practice of security testing. Garousi et al.[2] reported that security testing is least widely used among all kinds of testing (e.g., unit testing, integration testing). Sung et al.[42] found that in the New Zealand software industry, security testing and recovery testing practices are low compared to functional testing. The scenario is the same for Bangladesh; we found that 16.67% of respondents reported security testing to ensure security.

RQ2-D4. Security and performance measures used. Compared to other emerging SE industries, Bangladesh has less practice of no security. However, we have noticed a lack of testing practice to ensure security and performance. Besides, the Bangladesh SE industry lags in using new technologies (e.g., Container, Cloud).

5. Discussions

We analyze our survey responses and compare those with findings from other countries (when possible) across three participant demographics: profession (Section 5.1), experience (Section 5.2), and gender (Section 5.3).

5.1. Analysis by Professions

In Tables 9, 10, and 11, we summarize the interesting results of the closed questions by the reported roles of our survey participants. For all questions, we reported the top values for each role. For example, for the question 'Which of the following do you use for requirements gathering?' in Table 9, we reported Plain Text for the developer's role. 26.2% of software engineers use plain text for requirement gathering. Of all the other processes used by developers, the most commonly used requirement gathering process is plain text.

Agile is the most practiced requirement gathering method. The popularity of the Agile method is consistent across the different reported roles in our surveys. The second popular method is Scrum.

In Q8, participants were asked to identify the SDLC activity, where most of the time is spent. Generally, it is expected that participants in the senior role (e.g., manager, team lead) will spend time in requirement analysis, documentation where participants in the junior role (e.g., developers, r&d engineer) will spend time in implementation and testing. From Q8 in Table 9, we observe that senior role participants mostly spend time in requirement analysis and documentation. However, testing is not in the top three most times spent activity. For participants in the junior role, the most time-consuming activity is implementation, and the second-most time-consuming activity includes several other SDLC steps (e.g., maintenance, requirement analysis, etc.). This may indicate that in Bangladesh, SE industry testing is considered comparatively less important. **Across all roles, implementation is the most time spent activity in SDLC. However, testing is not considered one of the most time-consuming activities. Rigorous testing practices may not be prevalent in the Bangladesh SE industry.** In most of the roles, Java is the most used languages (Java is the second most used language in cases where JavaScript is the most used language). Even in the data engineer role, Java is the most used language, though Python is mostly used in data processing. Like Java, Spring (one framework of Java) is the widely used framework regardless of role. **Java and JavaScript are the most used languages regardless**

Table 9: Highlights of Findings from Survey Closed Questions by Profession

No	Question
4	For how many years have you coded professionally? 1) Architect: 1) Architect: more than 10 (100%) 2) Business analyst: more than 10 (100%) 3) Data Engineer: more than 10 (100%) 4) Developer: less than 2 (41.41%) 5) Manager: more than 10 (58.33%) 6) R&D: less than 2 (100%) 7) SQA: 2 to 5 (50%) 8) Team Lead: 5 to 10 (100%) 9) Trainer: 2 to 5 (100%) 10) UXD: 5 to 10 (100%)
6	Which of the following software development methodologies do you follow? 1) Architect: Agile (100%) 2) Business analyst: Agile (50%) 3) Data Engineer: Scrum (100%) 4) Developer: Agile (60.82%) 5) Manager: Agile (90%) 6) R&D: Agile (50%) 7) SQA: Agile (87.5%) 8) Team Lead: Agile (100%) 9) Trainer: Agile (100%)
7	Which of the followings do you use for requirements gathering? 1) Business analyst: GUI prototype (22.22%) 2) Data Engineer: Plain Text (33.33%) 3) Developer: Plain Text (26.2%) 4) Manager: Grooming seasons (20.48%) 5) R&D: Grooming seasons (33.33%) 6) SQA: Use Case (27.78%) 7) Team Lead: GUI prototype (33.33%) 8) Trainer: GUI prototype (25%) 9) UXD: GUI prototype (50%)
8	On which software development activities, do you spend most of the time? 1) Architect: Implementation (25%) 2) Business analyst: Implementation (33.33%) 3) Data Engineer: Implementation (33.33%) 4) Developer: Implementation (30.33%) 5) Manager: Requirement Analysis (22.37%) 6) R&D: Implementation (50%) 7) SQA: Testing (25%) 8) Team Lead: Documentation (25%) 9) Trainer: Implementation (25%) 10) UXD: Implementation (33.33%)

Table 10: Highlights of Findings from Survey Closed Questions by Profession

No	Question
9	Which of the following technologies do you have experience working in? 1) Architect: Mobile (50%) 2) Business analyst: Desktop (33.33%) 3) Data Engineer: Others (100%) 4) Developer: Web (49.39%) 5) Manager: Web (36.92%) 6) R&D: Web (66.67%) 7) SQA: Web (40.91%) 8) Team Lead: Desktop (50%) 9) Trainer: Desktop (33.33%) 10) UXD: Web (100%)
10	What is the primary operating system you are developing on? 1) Architect: Linux (66.67%) 2) Business analyst: Windows (66.67%) 3) Data Engineer: MacOS (100%) 4) Developer: Linux (42.86%) 5) Manager: Linux (53.49%) 6) R&D: Linux (50%) 7) SQA: Windows (56.25%) 8) Team Lead: Windows (100%) 9) Trainer: Others (100%) 10) UXD: Linux (50%)
11	Which programming languages are you using? 1) Architect: Java (100%) 2) Business analyst: JavaScript (40%) 3) Data Engineer: Go (25%) 4) Developer: JavaScript (31.75%) 5) Manager: JavaScript (27.06%) 6) R&D: Java (66.67%) 7) SQA: Java (24%) 8) Team Lead: C# (25%) 9) Trainer: C/C++ (25%) 10) UXD: JavaScript (100%)
12	Which frameworks are you using? 1) Architect: Spring (100%) 2) Business analyst: ASP. NET (16.67%) 3) Developer: Spring (26.61%) 4) Manager: Spring (23.73%) 5) R&D: Spring (66.67%) 6) SQA: ASP. NET (25%) 7) Team Lead: ASP. NET (50%) 8) Trainer: Django (100%)

of role. From Q15 of Table 11, we observe that developers mostly practice the highest automated testing level. In Q13 of Table 11 shows that developers mostly practice unit testing. It is comparatively easier to achieve automated testing through unit testing because of several testing frameworks. This may be

Table 11: Highlights of Findings from Survey Closed Questions by Profession

13	What types of software testing practices do you use? 1) Architect: Functional testing (25%) 2) Business analyst: Unit testing (40%) 3) Data Engineer: Performance testing (50%) 4) Developer: Unit testing (27.81%) 5) Manager: Unit testing (23.33%) 6) R&D: Performance testing (50%) 7) SQA: Functional testing (20.93%) 8) Team Lead: Functional testing (33.33%) 9) Trainer: Functional testing (25%)
15	What is the level of automated testing in your projects? 1) Architect: 3 (100%) 2) Business analyst: 1 (50%) 3) Data Engineer: 2 (100%) 4) Developer: 5 (40.66%) 5) Manager: 2 (36.36%) 6) R&D: 3 (50%) 7) SQA: 3 (44.44%) 8) Team Lead: 4 (100%) 9) Trainer: 4 (100%) 10) UXD: 4 (100%)
16	Which tools do you use for testing and quality assurance? 1) Architect: Selenium (50%) 2) Business analyst: JenKins (33.33%) 3) Data Engineer: XUnit (100%) 4) Developer: XUnit (35.14%) 5) Manager: XUnit (34.88%) 6) R&D: Selenium (50%) 7) SQA: Selenium (43.75%) 8) Team Lead: Selenium (50%) 9) Trainer: JenKins (100%)
17	Which tools do you use for continous deployment? 1) Business analyst: Jenkins (100%) 2) Developer: AWS codeDeploy (25.45%) 3) Manager: Jenkins (31.03%) 4) R&D: Open Source server (100%) 5) SQA: Bamboo (60%) 6) UXD: AWS codeDeploy (100%)

one of the reasons behind the high level of automated testing among developers. Automated tests offering libraries/frameworks for other types of testing can encourage higher automated testing levels in other roles. **Developers mostly practice the highest level of automated testing. Automated testing frameworks for other types of testings may increase the practice of automated testing.**

5.2. Analysis by Experience

We highlight the results of our survey's closed questions by the reported professional experiences of the survey respondents in Tables 12 and 13. The similar principles discussed in Section 5.1 are applied for this report.

With the increase of professional experience, we see the percentage of software developer being decreased but manager being increased according to Q5 in Table 12. On top of that, from Q7, we see to gather requirements of a software project, employees up-to mid-senior level, most of whom are developers, tend to use plain text, where more than 5 years experienced professionals prefer storyboard.

From the Q8 in Table 12, we see **implementation among all other development activities being the main concern for all levels of experienced employees. However, the ratio of the top two activities points out that the more senior an employee is, the more he/she tends to analyze the requirements of a software project.**

As per the Q10 in Table 12, we find that at the initial stage of career, professionals are inclined to prefer Windows most and then they mostly use Linux in mid-career and gradually they tend to use macOS at late-career. It might indicate that employees were proficient in Windows most before the start of their career. In Q5 of as per the Q10 in Table 12, we see the percentage of developer up-to mid-career is the most and in late-career, the percentage of managers is noticeable which point out that developers are inclined to use Windows and Linux, and managers prefer macOS for their managerial tasks.

When participants were asked about their software testing practices, most of them per experience level were concerned about the unit and functional testing, pictured in Q14 of Table 13. Moreover, we have found that senior participants are more prone to making software products accepted among clients using user acceptance testing (UAT). However, if we looked at Q15 when asked to level the automatic testing of their projects, the percentage is gradually decreasing with the seniority level like the ratio of the development activities of implementation and requirement analysis discussed earlier in this section. And most of the

Table 12: Highlights of Findings from Survey Closed Questions by Experience

<table>
<tr><th>No</th><th>Question</th></tr>
<tr><td>5</td><td>What is your current role?
1) less than 2: Developer (91.11%) 2) 2 to 5: Developer (75.0%) 3) 5 to 10: Developer (60.71%) 4) more than 10: Manager (42.42%) & Developer (42.42%)</td></tr>
<tr><td>6</td><td>Which of the following software development methodologies do you follow?
1) less than 2: Agile (55.74%) 2) 2 to 5: Scrum (34.78%) 3) 5 to 10: Agile (50.0%) 4) more than 10: Agile (33.33%)</td></tr>
<tr><td>7</td><td>Which of the followings do you use for requirements gathering?
1) less than 2: Plain Text (30.88%) 2) 2 to 5: Plain Text (29.03%) 3) 5 to 10: Story board (23.73%) 4) more than 10: Story board (27.63%)</td></tr>
<tr><td>8</td><td>On which software development activities, do you spend most of the time?
Implementation = Imp, Requirement Analysis = RA
<table>
<tr><th></th><th>top1 (%)</th><th>top2 (%)</th><th>top1 / top2</th></tr>
<tr><td>less than 2</td><td>Imp (29.9)</td><td>RA (19.59)</td><td>1.53</td></tr>
<tr><td>2 to 5</td><td>Imp (32.88)</td><td>RA (15.07)</td><td>2.18</td></tr>
<tr><td>5 to 10</td><td>Imp (21.69)</td><td>RA (19.28)</td><td>1.125</td></tr>
<tr><td>more than 10</td><td>Imp (23.68)</td><td>RA (23.68)</td><td>1</td></tr>
</table></td></tr>
<tr><td>9</td><td>Which of the following technologies do you have experience working in?
1) less than 2: Web (51.43%) 2) 2 to 5: Web (56.0%) 3) 5 to 10: Web (46.0%) 4) more than 10: Web (35.38%)</td></tr>
<tr><td>10</td><td>What is the primary operating system you are developing on?
1) less than 2: Windows (46.15%) & Linux (30.77%), 2) 2 to 5: Linux (47.73%) & Windows (38.64%), 3) 5 to 10: Linux (41.46%) & Windows (26.83%), 4) more than 10: Linux (52.27%) & MacOS (25.0%)</td></tr>
</table>

Table 13: Highlights of Findings from Survey Closed Questions by Experience

No	Question
11	Which programming languages are you using? 1) less than 2: Java (31.11%) & JavaScript (30.0%), 2) 2 to 5: Java (28.57%) & JavaScript (28.57%), 3) 5 to 10: JavaScript (28.17%) & Java (23.94%), 4) more than 10: JavaScript (27.03%) & Java (24.32%)
12	Which frameworks are you using? 1) less than 2: Spring (35.56%), 2) 2 to 5: Others (25.58%) & Spring (18.6%), 3) 5 to 10: Spring (29.55%), 4) more than 10: Spring (27.78%)
14	What types of software testing practices do you use? (Functional testing = FT, Performance testing = PT, Unit testing = UT, User acceptance testing = UAT) 1) less than 2: UT (32.1%) & FT (24.69%), 2) 2 to 5: FT (24.64%) & UT (18.84%), 3) 5 to 10: FT (23.44%) & UAT (20.31%), 4) more than 10: UT (25.29%) & UAT (19.54%)
15	What is the level of automated testing in your projects? 1) less than 2: 5, 34.88% 2) 2 to 5: 5, 35.71% 3) 5 to 10: 5, 30.77% 4) more than 10: 2, 27.27%
16	Which tools do you use for testing and quality assurance? 1) less than 2: XUnit (33.33%), 2) 2 to 5: Selenium (45.45%), 3) 5 to 10: Selenium (35.48%), 4) more than 10: XUnit (38.46%)
17	Which tools do you use for continous deployment? (AWS codeDeploy = AC) 1) less than 2: AC (23.53%), 2) 2 to 5: AC (26.67%) & Bamboo (20.0%), 3) 5 to 10: AC (27.27%), 4) more than 10: Jenkins (37.5%) & AC (16.67%)

more than ten years of experience participants leveled test automation ‘2’ for their projects. It implies that when it’s time for testing practices like user acceptance testing rather than unit and functional testing, software industries are less likely to use test automation tools in their projects. **Testing practices like unit testing, functional testing, etc., used for implementation purposes, are mainly carried out as test automation in Bangladesh. But, for other testing practices like user acceptance testing, software companies tend not to carry out test automation tools more often.**

At Q17 of Table 13 we observe that using tools for continuous deployment is commensurate to the years of professional experience. Personnel of senior-level are more likely to work in development and operations (DevOps) related fields.

5.3. Analysis by Gender

In our survey, 90.1% of participants were male, and 9.9% participants were female, which is slightly better than the Stack Overflow (SO) survey [36, 31, 20, 19] (in Stack Overflow 8% respondents marked them as female). It is often said that females are less represented in STEM. As the SE industry is directly related to STEM, the claim may be true in the SE industry. Though our survey does not represent the real scenario, the proportion of male and female participants supports the claim of under-representation. To get an overview of the Bangladesh SE industry by gender in Figure 21, we plotted the participants' roles grouped by gender. **In terms of**

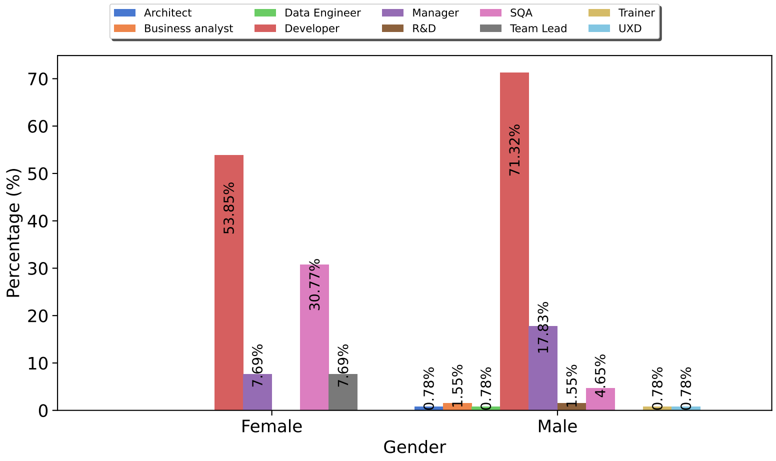

Figure 21: Gender based role

roles, the proportion of developers among female participants is comparatively low than that of male participants. However, the scenario is the opposite of the manager and team lead role. Generally, the developer role is considered a junior role, and the manager/team lead is considered a senior role. Thus, it is clear from Figure 21 that there is a difference between junior and senior roles among female participants. This indicates that in recent years females are less interested in joining the SE industry. **Also, among ten different roles, female participants are holding only four types of roles. Our findings align with the survey result of Hussain et al.[13]. They found that female participants are only limited to developer, QA, and project manager roles, where male participants are holding varying types of roles in the Bangladesh SE industry. The result of the Stack Overflow survey[19] is slightly different. In the SO survey, female respondents are mostly data scientists, business analysts, QA, and developer roles. A common theme in all surveys is the role of female respondents in the QA.**

The interesting findings of gender analysis are presented in Table 14. James et al.'s [48] survey on male and female software professionals found that men are more likely to be in senior positions than women. In Q4 of Table 14, we observe a similar scenario. However, in our case, the observation is not statistically significant. **James et al. [48] also found that male software practitioners tend to be older than female practitioners and female practitioners tend to leave jobs in mid-career. We observe the same phenomena in Q2 and Q4, and the finding may be true in Bangladesh.**

James et al.[48] found that female practitioners express less satisfaction with the spirit of teamwork inside the organization. They conclude that this characteristic would make male practitioners a key part of a good agile team. Our findings differs from James et al. [48]. In Q6 of Table 14, we observe that our survey's female participants prefer the Agile methodology over other software development methodologies. However, development methodology selections are often performed by senior management. Thus, personal preferences can have

Table 14: Highlights of Findings from Survey Closed Questions by Gender

No	Question
2	What is your age? 1) 15 to 20: Male (100%) 2) 20 to 25: Male (87.8%) 3) 25 to 30: Male (86.84%) 4) 30 to 35: Male (96%) 5) 35 to 40: Male (100%) 6) 40 to 45: Male (100%)
4	For how many years have you coded professionally? 1) less than 2: Male (88.89%) 2) 2 to 5: Male (91.67%) 3) 5 to 10: Male (82.14%) 4) more than 10: Male (100%)
5	What is your current role? 1) Architect: Male (100%) 2) Business analyst: Male (100%) 3) Data Engineer: Male (100%) 4) Developer: Male (92.93%) 5) Manager: Male (95.83%) 6) R&D: Male (100%) 7) SQA: Male (60%) 8) Team Lead: Female (100%) 9) Trainer: Male (100%) 10) UXD: Male (100%)
6	Which of the following software development methodologies do you follow? 1) Agile: Male (86.36%) 2) Kanban: Male (100%) 3) Others: Male (100%) 4) Pair Programming: Male (96.43%) 5) Scrum: Male (93.75%) 6) Waterfall: Male (100%) 7) XP: Female (100%)
9	Which of the following technologies do you have experience working in? 1) Cloud: Male (100%) 2) Desktop: Male (78.57%) 3) Embedded /IOT: Male (100%) 4) Mobile: Male (90.62%) 5) Others: Male (100%) 6) Web: Male (90%)

minimal effect on this choice

In James et al. [48] survey, neither male nor female was not dominating any particular technology field. However, in Q9 of Table 14 we observe female participants working on desktop, web, and mobile. As a small SE industry, cloud/IoT is considered a less secured field as job opportunities in these fields are small. It seems female participants tend to work in a more secure technology (in terms of a job opportunity) field.

6. Implications of Findings

The findings from our study can guide the following major stakeholders in software engineering (SE): (1) SE Tool Creators to develop usable and affordable automated testing framework that can be accessible to emerging countries, (2) SE Researchers to compare and contrast software development practices in emerging countries with respect to region-specific and global trends, (3) SE career enthusiasts who would like to participate in the high-growth software industries in the emerging countries, (4) SE security and performance practitioners to develop techniques to better enforce such crucial non-functional requirements in software products in emerging countries, and (5) SE Industry leaders to offer customized region-specific software products and to promote diversity irrespective of regions. We discuss the implications below.

Tool Creators. We have found that rigorous testing practice is not prevalent in Bangladesh. The difference in testing effort between the established software industry (e.g., Canada) and Bangladesh is too high. The scenario is also true for other developing countries like Pakistan. From our study, software practitioners may have an idea of their standing in software QA in testing. Similar to all emerging industries, security testing is less prioritized in Bangladesh. From our comparison, developers may better understand security testing and security practices in other countries. While these findings show a difference in the adoption of testing tools and non-functional measures between emerging and developed countries, the reasons could be multi-faceted. For example, one reason of the prevalence of less automated testing in emerging countries like Bangladesh is that most of the developed software products are web-based. This means that developers in Bangladesh need to test their GUI-based software product. Proper software testing tool support for GUI testing is in infancy compared to the testing of source code. Good GUI-based testing tools are also not free or open-source. This makes it hard for developers in emerging countries to learn and use GUI-based testing framework. Therefore, while unit testing is practiced widely in Bangladesh as like any other countries, the lack of automated testing is really due to the nature of the software products they developers are entitled to work on and the availability of good, usable and affordable testing framework to support them. Therefore, SE tool creators may create a user-friendly framework to implement automatic testing for different types of testing to address this affordability issue. Alternatively, cheaper and/or more affordable versions of the industry standard automated testing framework can be offered to developers of emerging countries. Containerization technologies are a popular trend worldwide. However, the practice of containerization technologies is too low. Automatic deployment/release is another under-practiced area. Tool creators can investigate the challenges to incorporate these technologies in an emerging SE industry like Bangladesh and offer more affordable alternatives. This is important because a software product developed in an emerging country like Bangladesh is actually consumed mostly in the developed countries (e.g., via outsourcing). Therefore, anyone can suffer from any lingering faults in the developed software products.

Researchers. In SE research, we need to be aware of the current trends in software development practices not only to guide our research along the trends but also to ensure that our research contributions are timely and effective to the current needs of the software industry. In research, we strive to achieve generalizability of our findings, because otherwise we run the risk of becoming too niche or specific that may not cater to a global audience. Our study results of software development practices in an emerging country like Bangladesh and the comparison of such practices against multiple countries worldwide show that development practices, tools, techniques all could vary across the countries. Therefore, it may not be always possible to observe or enforce similar development practices across the globe, even when certain practices may be perceived as superior to others. A major reason of the observed differences between the emerging countries and developed countries is that outsourcing is a still a major source of revenue for the software industries in emerging countries.

This explains why product design and overall architecture design is not widely practised in the software industries of emerging countries. A major focus in software engineering research is to understand team cohesion and collaboration. Given the outsourcing nature of the works carried by the software companies in the emerging countries, we can expect that the developers in the companies need to adapt to constantly dynamic teams where some team members are local and some others are remote and temporary, e.g., members from a software company of a developed country that are engaged in the outsourcing work. It is important to understand the productivity of the developers in such dynamic environment, the challenges they face (e.g., lack of enough in-person communication due to remote work and time difference). Such research can offer findings in the development new methodologies, tools and techniques to facilitate better communication and collaboration practices among the developers in the such teams. This can then also help the software companies in emerging countries to offer better support to the overall system design and planning.

Career Enthusiasts. We have found that certain languages (e.g., Java, JavaScript, etc.) and frameworks (e.g., Spring, Django, ASP.NET) have extensive use in the software industry of Bangladesh. This finding is not different from other emerging countries. Therefore, career enthusiasts who aspire for a software development profession in emerging countries may focus on mastering such skills. Universities can update their curricula to meet industry demand. The students aspiring to join the software industry must prepare themselves accordingly to be productive quickly. We have also found less automated testing practices in the industry of Bangladesh. This may be related to a lack of exposure to the testing framework. Similar to Hussain et al.[13], we also suggest including testing-related courses into the curriculum of universities and introducing relevant assignments to have hands-on experience on automated testing tools right from the student level. The students in their academic projects should also use container and other DevOps tools to bring some qualitative improvement for the industry when they join there.

Security and Performance Practitioners. More than 9.5% of our respondents responded that they do not take any measures to mitigate the security risk. The common reason for not taking any measures is (1) the product is an early stage (2) the respondents' role does not require any product security measures. The reason for not taking any security measures is different from North America [49]. The reasons are (1) there are no formal test plans, (2) lack of knowledge regarding testing tools. Though respondents do not think about product security initially, it is recommended [50, 51] to plan security tests and product security from the design phase.

Modern frameworks provide the basic security of the solution. Moreover, some framework provides enhanced, focused, customized security through a plug-in or add-on, e.g., spring-security, spring-cloud security. Framework-based security is a growing practice in the software industry[52]. The practice in the Bangladesh SE industry matches the global practice. According to our survey, it is the second most popular measure to mitigate security risk. Survey respondents have reported using OWASP, HDIV, and spring security. Srinivasan et al.[53] have conducted a comparison among the popular web frameworks based on security. Based on five criteria, they ranked the frameworks, and all of the mentioned frameworks of our respondents are in the top 10 list. It seems that secure software engineering practice is prevalent in the Bangladesh SE industry.

On a survey of 237 software professionals, Elahi et al.[54] found that 51% of respondents maintain at least one security standard, and 19% of respondents maintain ISO 17799 security standard. On the contrary, about 29.63% of respondents of the Bangladesh software industry maintain any security standard. It is clear that security standards are not that much popular in this SE industry.

According to Smith et al.[55], efficient architecture and continuous monitoring tools are two of the twenty-four best practices to ensure software performance. The respondents report both practices. However, Smith et al. presented twenty-one other best practices, and we have not found other practices in our survey. It is clear the SE industry of Bangladesh only practices a few best measures for ensuring software performance.

Bondi et al.[56] have listed four scalability types to ensure software capability to scale; however, we observe only one scalability type (load scalability) in our responses. To obtain software scalability use of cloud services is one of the most popular strategies[57]. Another popular strategy is the use of microservice. Microservice and cloud services together allow the user to scale up and down any system dynamically. Cáceres et al.[58] reported that cloud services and microservice-based architecture are generally used together to ensure scalability. In the Bangladesh SE industry, this practice may be prevalent. The use of cloud service and efficient use of architecture are the second and third most popular topic among our respondents.

Overall, the findings show that much need to be done to ensure proper security and performance measures in the software development practices of emerging countries. Multi-faceted efforts are warranted like development of affordable tools and techniques for the emerging countries, enforcement of widely-accepted and measurable industry standards across the regions, and the proper training of the security and performance principles to the developers in emerging countries.

Industry Leaders. This study found that the Bangladesh software industry lags in adopting some of the current industry trends. Bangladesh can be a good marketplace for cloud companies like Microsoft, Google, and Amazon if they provide an affordable package for the software companies operating here. Local entrepreneurs may also think of building cost-effective local public clouds along with providing some common DevOps services.

Moreover, we have found most of the participants in our study are engaged in web technology. This may indicate that the web is the most popular form among users of Bangladesh. This observation can help the industry owners to select appropriate medium while targeting users of this region. common practices of the practitioners of the Bangladesh software industry.

Finally, we have identified that the participation of female in the Bangladesh SE industry is comparatively less than that of male engineers (see Section 5.3). Hussain et al.[13] expressed a concern that there may be bias in the hiring process of the industry. Overall, the under-representation of females and minorities in software industries worldwide is a prevalent and ongoing concern. This was also reported in the 2020 Stack Overflow developer survey, where more than 90% respondents were males. Our findings confirm similar trends in Bangladesh. Therefore, proper measures need to be taken to encourage equity, diversity, and inclusion in software industries across the regions. Such measures may be better enforced by tailoring the measures to region-specific cultural attributes.

7. Threats to Validity

We discuss the threats to validity of our studies following common guidelines for empirical studies [59].

Construct validity is mainly concerned with the extent to which the study objectives truly represent the theory behind the study [60]. In our study, we have used open coding strategy to label the survey responses. The nature of this coding strategy may introduce researcher bias into coded labels. To mitigate the issue, the labels have been coded by two individuals, and the codes are accepted when there is a reasonable agreement among the coders. Another issue can be whether our data actually represents real-world SE practices. This study counted the votes and made statistical inference, which is common in survey-based studies. It is believed that voting data can, to a certain extent, reflect the opinions of the majority. It was previously observed[2] that people tend to form their answers close to expected answers when evaluated. To mitigate the threat, before the survey, we informed participants that our motive in this survey was to get a decent understanding of current practices, and we do not intend to collect any personally identifiable data. Construct threats may also be introduced by a misleading interpretation of the survey questions. We conducted a preliminary survey and interview session with some participants to rule out any ambiguity from survey questions and thus tried to reduce such risk.

Internal validity is a property of scientific studies that refers to how well a study has been conducted. A threat to internal validity in this study is inherent in the participant selection bias. We used several social platforms, personal connections to reach as many participants as possible. Another threat could arise from the placement of the options in a multiple-choice question. It is often observed that survey participants often show bias towards the first option in any multiple-choice question[61]. However, in one of the multiple-choice questions (Q9), the 'Web' option was placed at the bottom of the list. Despite this placement, we observed most of the participants selected 'Web' as a technology platform. But practically, from the personal experience of the authors, there is no bias in this opinion.

External validity is concerned with the generalization of the study result. In our study, we have participants from almost all the groups of the Bangladesh software industry. However, it is difficult to claim the statistical generalizability of our findings, given that our sample included 137 respondents where there 1100+ companies and 3,00,000 IT professionals[6] in the software industry of Bangladesh. Moreover, emerging IT industries share a common trend of challenges[11, 62]. Thus, our findings are also applicable to other emerging software industries across the globe.

8. Conclusion

This study identifies the general practices, obstacles, and requirements of the software development industry in Bangladesh. Here we explored the common practices and shortcomings in the emerging industry. It was revealed that testing and automated testing are quite low compared to the established software industry. Although we have found that software security consciousness is higher than other emerging software industries, security testing practices, and standards lag behind other countries. We have noticed the low participation of women in the software industry. Further research may be conducted to determine any biases or barriers for women in the SE industry. These findings can help developers, industry owners, and academia work to better the industry from their own position and perspective.

References

[1] V. Garousi, J. Zhi, A survey of software testing practices in canada, Journal of Systems and Software 86 (5) (2013) 1354–1376.

[2] V. Garousi, A. Cokunay, A. Betin-Can, O. Demirrs, A survey of software engineering practices in turkey, The Journal of Systems and Software 108 (2015) 148–177.

[3] F. Vonken, J. Brunekreef, A. Zaidman, F. Peeters, Software Engineering in the Netherlands: The State of the Practice, Tech. rep., Software Engineering Research Group, Department of Software Technology, Delft University of Technology (12 2012).

[4] D. Wang, M. Galster, Development processes and practices in a small but growing software industry a practitioner survey in new zealand, in: Proceedings of ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Oulu, Finland, October 1112, 2018 (ESEM18), ACM Press, 2018, pp. 1–10.

[5] F. Baharom, A. Deraman, A. R. Hamdan, A survey on the current practices of software development process in malaysia, in: 17] NOAA Coastal Services Center, "Introduction to Survey Design and Delivery," Charleston: NOAA Coastal Services, 2006, pp. 57–76.

[6] Basis, It and ites industry overview (2018).
URL https://basis.org.bd/public/files/publication/5e127d7cb6967ba96136a3b168568073f9800e5b0f5b9.pdf

[7] M. Cusumano, A. MacCormack, C. Kemerer, B. Crandall, Software development worldwide: The state of the practice, IEEE Software 20 (6) (2003) 28–34.

[8] A. AlSubaihin, F. Sarro, S. Black, L. Capra, M. Harman, App store effects on software engineering practices, IEEE Transactions on Software Engineering (2019) 1–1.

[9] E. Klotins, M. Unterkalmsteiner, T. Gorschek, Software engineering in start-up companies: An analysis of 88 experience reports, Empirical Software Engineering 24 (1) (2018) 68–102.

[10] L. Groves, R. Nickson, G. Reeve, S. Reeves, M. Utting, A survey of software development practices in the new zealand software industry, in: Proceedings of Australian Software Engineering Conference (ASWEC), 2000.

[11] R. Sison, S. Jarzabek, O. Hock, W. Rivepiboon, N. Hai, Software practices in five asean countries: an exploratory study, in: Proceeding of the 28th International Conference on Software Engineering (ICSE), 2006.

[12] I. Zafar, A. Shaheen, A. K. Nazir, B. Maqbool, W. H. Butt, J. Zeb, Why pakistani software companies don't use best practices for requirement engineering processes, in: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE, 2018.

[13] I. Hussain, A. S. Hasan, R. M. Yasir, A. Kabir, S. I. Ahmed, How well does undergraduate education prepare software engineers? perspectives of practitioners in bangladesh, in: 2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T), IEEE, 2020.

[14] M. S. Rahim, M. H. Hasana, A. E. Chowdhury, S. Das, Software engineering practices and challenges in bangladesh: A preliminary survey, Journal of Telecommunication 9 (3) (2017) 163–169.

[15] S. M., M. Shamsur, A. Z., M. Hasibul, A survey of software quality assurance and testing practices and challenges in bangladesh, International Journal of Computer Applications 180 (39) (2018) 1–8.

[16] Z. Begum, M. S. A. Khan, M. Hafiz, M. S. Islam, M. Shoyaib, Sofware development standard and software engineering practice: A case study of bangladesh, Journal of Bangladesh Academy of Sciences 32 (2) (Jan. 2009).

[17] W. Vogt, R. B. Johnson, Dictionary of Statistics and Methodology - A Nontechnical Guide for the Social Sciences, Sage, 2005.

[18] J. W. Creswell, Research design: Qualitative, quantitative, and mixed methods approaches, Sage, 2013.

[19] Stackoverflow, Stackoverflow Survey 2020 (2020).
URL https://insights.stackoverflow.com/survey/2020

[20] Stackoverflow, Stackoverflow Survey 2019 (2019).
URL https://insights.stackoverflow.com/survey/2019

[21] Basis, Basis overview (2019).
URL https://basis.org.bd/public/files/publication/5e12287c79d72b9e180fbeb8cd623d1a657be96c4e208.pdf

[22] C.-J. Lu, S. W. Shulman, Rigor and flexibility in computer-based qualitative research: Introducing the coding analysis toolkit, International Journal of Multiple

Research Approaches 2 (1) (2008) 105–117.

[23] D. Freelon, Recal2: Reliability for 2 coders (2020).
URL http://dfreelon.org/utils/recalfront/recal2/

[24] J. Cohen, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1) (1960) 37–46.

[25] W. A. Scott, Reliability of content analysis: The case of nominal scale coding, The Public Opinion Quarterly 19 (3) (1955) 321–325.

[26] k. krippendorff, Reliability in content analysis: Some common misconceptions and recommendations, Human Communication Research 30 (2004) 411–433.

[27] M. Joyce, Picking the best intercoder reliability statistic for your digital activism content analysis, 2013.

[28] J. R. Landis, G. G. Koch, The measurement of observer agreement for categorical data, Biometrics 33 (1) (1977) 159.

[29] S. Seabold, J. Perktold, statsmodels: Econometric and statistical modeling with python, in: 9th Python in Science Conference, 2010.

[30] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272.

[31] Stackoverflow, Stackoverflow Survey 2018 (2018).
URL https://insights.stackoverflow.com/survey/2018

[32] H. Cramer, Mathematical methods of statistics / by Harald Cramer, 1946.

[33] D. J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, 4th Edition, Chapman ; Hall/CRC, 2007.

[34] JetBrains, The State of Developer Ecosystem 2020 (2020).
URL https://www.jetbrains.com/lp/devecosystem-2020/

[35] Github, The State of the Octoverse.
URL https://octoverse.github.com/

[36] Stackoverflow, Stackoverflow Survey 2017 (2017).

URL https://insights.stackoverflow.com/survey/2017

[37] JetBrains, The State of Developer Ecosystem 2019 (2019).
URL https://www.jetbrains.com/lp/devecosystem-2019/

[38] M. A. T. Almomani, S. Basri, A. K. B. Mahmood, A. O. Bajeh, Software development practices and problems in malaysian small and medium software enterprises: A pilot study, in: 2015 5th International Conference on IT Convergence and Security (ICITCS), IEEE, 2015.

[39] S. Dutta, M. Lee, L. V. Wassenhove, Software engineering in europe: A study of best practices (1999).

[40] S. Farvin, F. Baharom, A. Deraman, J. Yahaya, H. Haslina, An exploratory study on secure software practices among software practitioners in malaysia 8 (2016) 39–45.

[41] S. Bahl, O. P. Wali, P. Kumaraguru, Information security practices followed in the indian software services industry: An exploratory study, in: 2011 Second Worldwide Cybersecurity Summit (WCS), 2011, pp. 1–7.

[42] P. Sung, B. Sung, J. Paynter, Software testing practices in new zealand (01 2006).

[43] C. Phillips, J. Alam, Software engineering practices and tool support: an exploratory study in new zealand, Australasian Journal of Information Systems 11 (1) (Nov. 2003).

[44] M. S. Jahan, M. T. Riaz, Kashif, M. Abbas, Software testing practices in IT industry of pakistan, in: Proceedings of the 6th Conference on the Engineering of Computer Based Systems - ECBS '19, ACM Press, 2019.

[45] P. Laihonen, Adoption of devops practices in the finnish software industry: an empirical study, Master's thesis (2018).

[46] W. Hussain, T. Clear, S. MacDonell, Emerging trends for global DevOps: A new zealand perspective, in: 2017 IEEE 12th International Conference on Global Software Engineering (ICGSE), IEEE, 2017.

[47] Iqra Basharat, Mamuna Fatima, R. Nisa, R. Hashim, A. Khanum, Requirements engineering practices in small and medium software companies: An empirical study (2013).

[48] T. James, M. Galster, K. Blincoe, G. Miller, What is the perception of female and male software professionals on performance, team dynamics and job satisfaction? insights from the trenches, in: 2017 IEEE/ACM 39th International Conference

on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), IEEE, 2017.

[49] H. Assal, S. Chiasson, 'think secure from the beginning', in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19, ACM Press, 2019.

[50] S. Chandra, R. A. Khan, A. Agrawal, Software security factors in design phase, in: Information Systems, Technology and Management, Springer Berlin Heidelberg, 2009, pp. 339–340.

[51] Z. Azham, I. Ghani, N. Ithnin, Security backlog in scrum security practices, in: 2011 Malaysian Conference in Software Engineering, IEEE, 2011.

[52] F. T. Alssir, M. Ahmed, Web security testing approaches: Comparison framework, in: Advances in Intelligent and Soft Computing, Springer Berlin Heidelberg, 2012, pp. 163–169.

[53] S. M. Srinivasan, R. S. Sangwan, Web app security: A comparison and categorization of testing frameworks, IEEE Software 34 (1) (2017) 99–102.

[54] G. Elahi, E. Yu, T. Li, L. Liu, Security requirements engineering in the wild: A survey of common practices, in: 2011 IEEE 35th Annual Computer Software and Applications Conference, IEEE, 2011.

[55] C. U. Smith, L. G. Williams, Best practices for software performance engineering, in: 29th International Computer Measurement Group Conference, December 7-12, 2003, Dallas, Texas, USA, Proceedings, Computer Measurement Group, 2003, pp. 83–92.

[56] A. B. Bondi, Characteristics of scalability and their impact on performance, in: Proceedings of the second international workshop on Software and performance - WOSP '00, ACM Press, 2000.

[57] J. Gao, P. Pattabhiraman, X. Bai, W. T. Tsai, SaaS performance and scalability evaluation in clouds, in: Proceedings of 2011 IEEE 6th International Symposium on Service Oriented System (SOSE), IEEE, 2011.

[58] J. Cáceres, L. M. Vaquero, L. Rodero-Merino, Á. Polo, J. J. Hierro, Service scalability over the cloud, in: Handbook of Cloud Computing, Springer US, 2010, pp. 357–377.

[59] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering: an introduction, Kluwer Academic Publishers,

Norwell, MA, USA, 2000.

[60] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, A. Wessln, Experimentation in Software Engineering, Springer Publishing Company, Incorporated, 2012.

[61] G. Uddin, O. Baysal, L. Guerrouj, F. Khomh, Understanding how and why developers seek and analyze API-related opinions, IEEE Transactions on Software Engineering (2019) 1–1.

[62] W. W. Lloyd, Automation challenges in southeast asia (2020).